

BASS_CONFIG_WMA_BASSFILE config option

Have BASS handle the reading of WMA files?

```
BASS_SetConfig(  
    BASS_CONFIG_WMA_BASSFILE,  
    BOOL bassfile  
);
```

Parameters

bassfile BASS should handle file reading?

Remarks

By default, BASSWMA will let the Windows Media modules handle the reading of WMA files, which disables some features. The *offset* and *length* [BASS_WMA_StreamCreateFile](#) (and [BASS_StreamCreateFile](#) via the plugin system) parameters are ignored. [BASS_StreamGetFilePosition](#) isn't fully supported. ID3 and ID3v2 tags aren't read (they shouldn't really be used in WMA files anyway).

Using this config option, BASS can be made to handle the WMA file reading instead, re-enabling the aforementioned features. There is a down side though, in that it's not possible to play WMA files while they are still being encoded.

BASS will only handle the reading of local WMA files. Internet files/streams will always be handled by the Windows Media modules, regardless of this config setting.

See also

[BASS_WMA_StreamCreateFile](#)

[BASS_GetConfig](#), [BASS_SetConfig](#)

BASS_CONFIG_WMA_PREBUF config option

Prebuffer internet streams during creation?

```
BASS_SetConfig(  
    BASS_CONFIG_WMA_PREBUF,  
    BOOL prebuf  
);
```

Parameters

prebuf Prebuffer internet streams on creation?

Remarks

The Windows Media modules must prebuffer a stream before starting decoding/playback of it. This option determines whether the stream creation function (eg. [BASS_WMA_StreamCreateFile](#)) will wait for the prebuffering to complete before returning. If playback of a stream is attempted before it has prebuffered, it will stall and then resume once it has finished prebuffering. The prebuffering progress can be monitored via [BASS_StreamGetFilePosition](#) (BASS_FILEPOS_WMA_BUFFER).

This option is disabled by default.

See also

[BASS_GetConfig](#), [BASS_SetConfig](#)

BASS_CONFIG_WMA_VIDEO config option

Play the audio from Windows Media Video (WMV) files?

```
BASS_SetConfig(  
    BASS_CONFIG_WMA_VIDEO,  
    BOOL video  
);
```

Parameters

video Accept WMV files?

Remarks

This option is enabled by default, and applies both when using [BASS_WMA_StreamCreateFile](#) and the plugin system.

See also

[BASS_WMA_StreamCreateFile](#)

[BASS_GetConfig](#), [BASS_SetConfig](#)

BASS_WMA_GetWMObject

Retrieves a pointer to the IWMReader interface of a WMA stream, or IWMWriter interface of a WMA encoder.

```
void *BASS_WMA_GetWMObject(  
    DWORD handle  
);
```

Parameters

handle The WMA stream or encoder handle.

Return value

If successful, then a pointer to the requested object is returned, otherwise NULL is returned. Use [BASS_ErrorGetCode](#) to get the error code.

Error codes

BASS_ERROR_HANDLE *handle* is not valid.

Remarks

This function allows those that are familiar with the Windows Media Format SDK to access the internal object interface, for extra functionality. If you create any objects through a retrieved interface, make sure you release the objects before calling [BASS_StreamFree](#) or [BASS_WMA_EncodeClose](#).

When streaming local (not internet) files, this function will usually actually return an `IWMSyncReader` interface instead of an `IWMReader` interface. The type of interface can be determined by querying other interfaces from it, eg. `IWMReaderAdvanced`.

See the Windows Media Format SDK for information on the `IWMReader`, `IWMWriter`, and associated interfaces.

BASS_WMA_StreamCreateFile

Creates a sample stream from a WMA file or URL.

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

Parameters

mem	TRUE = stream the file from memory, 2 = stream from an IStream object
file	Filename or URL (mem = FALSE) or a memory location (mem = TRUE) or a pointer to an IStream object (mem = 2).
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.
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 cannot be used together with this 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 DirectX 8 effects. See the DX8 effect implementations section for details. Use BASS_ChannelSetFX 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 BASS_ChannelGetData 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_SPEAKER_xxx	BASS_SAMPLE_FX flags are also ignored.
BASS_ASYNCFILE	<p><u>Speaker assignment flags</u>. These flags have no effect when the stream is more than stereo.</p> <p>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 <u>BASS_CONFIG_ASYNCFILE_BUFFER</u> config option. This flag only applies when the <u>BASS_CONFIG_WMA_BASSFI</u> config option is enabled, and is ignored when streaming from memory (<i>mem = TRUE</i>).</p>
BASS_UNICODE	<i>file</i> is in UTF-16 form. Otherwise it is ANSI.

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_WMA	The Windows Media modules (v9 or above) are not installed.
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	One or more of the parameters are invalid.
BASS_ERROR_FILEOPEN	The file could not be opened.
BASS_ERROR_FILEFORM	The file's format is not recognised/supported.
BASS_ERROR_CODEEC	There is no appropriate codec installed to decode the file. Try installing the latest Windows Media codecs.
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_WMA_LICENSE	The WMA file cannot be played because it is protected.
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 bitrate (amongst other things) can be retrieved through [BASS_ChannelGetTags](#) (BASS_TAG_WMA), which will return a pointer to a series of null-terminated UTF-8 strings, the final string ending with a double null. If the stream contains mid-stream tags (script), the latest tag can be retrieved through [BASS_ChannelGetTags](#) (BASS_TAG_WMA_META), which will return a single UTF-8 string. Each tag is in the form of "key=value".

A description of the codec used by the file is also available from [BASS_ChannelGetTags](#) (BASS_TAG_WMA_CODEC). 2 null-terminated UTF-8 strings are returned, with the 1st string being the name of the codec, and the 2nd containing additional information like what VBR setting was used.

The playback length of the stream can be retrieved using [BASS_ChannelGetLength](#). Until the whole file has been streamed, whatever length the file's header says is returned, which may or may not be exact.

Although the Windows Media modules uses its own internet streaming routines (not BASS's), the [BASS_CONFIG_NET_PROXY](#) and [BASS_CONFIG_NET_TIMEOUT](#) config options do have effect when opening WMA streams. When the [BASS_CONFIG_NET_PLAYLIST](#) config option is enabled, BASSWMA will process ASX and WPL files. None of the other NET config options apply.

Unless the [BASS_CONFIG_WMA_BASSFILE](#) config option is enabled, the Windows Media modules uses its own file reading routines, and the *offset* and *length* parameters are ignored, except that *length* is still the length when playing from memory. Also, [BASS_StreamGetFilePosition](#) is not fully supported. The file size (BASS_FILEPOS_END) can be retrieved, but the decode position (BASS_FILEPOS_CURRENT) is not available. The download progress of streamed files (BASS_FILEPOS_DOWNLOAD) can also be retrieved. The buffering progress (percentage) can be retrieved using the BASS_FILEPOS_WMA_BUFFER mode.

When streaming a file from the internet, it is not possible to seek with

[BASS_ChannelSetPosition](#) until the whole file has been downloaded. A sync (BASS_SYNC_DOWNLOAD) can be set to be notified when the file has been downloaded. When streaming from the internet, the WMA decoding is performed in a separate thread, so the CPU used to decode the stream during playback will not be included in the [BASS_GetCPU](#) return value.

The playback rate of local files can be altered with [BASS_ChannelSetAttribute](#). The playback rate of internet streams should not be changed, because they are delivered at a fixed rate: the rate required to sustain playback at normal speed. So increasing the rate will result in playback stalling.

See also

[BASS_WMA_StreamCreateFileAuth](#), [BASS_WMA_StreamCreateFileUser](#),
[BASS_CONFIG_WMA_BASSFILE](#), [BASS_CONFIG_WMA_PREBUF](#)

[BASS_ChannelGetInfo](#), [BASS_ChannelGetLength](#), [BASS_ChannelPlay](#),
[BASS_ChannelSetAttribute](#), [BASS_ChannelSetDSP](#), [BASS_ChannelSetFX](#),
[BASS_ChannelSetLink](#), [BASS_StreamFree](#), [BASS_ChannelGetTags](#)

BASS_WMA_StreamCreateFileAuth

Creates a sample stream from a WMA file or URL, optionally with a username and password to authenticate.

```
HSTREAM BASS_WMA_StreamCreateFileAuth(  
    BOOL mem,  
    void *file,  
    QWORD offset,  
    QWORD length,  
    DWORD flags,  
    char *user,  
    char *pass  
);
```

Parameters

mem	TRUE = stream the file from memory, 2 = stream from an IStream object
file	Filename or URL (mem = FALSE) or a memory location (mem = TRUE) or a pointer to an IStream object (mem = 2).
offset	Unused... set to 0.
length	Data length (only used if mem = TRUE).
flags	A combination of these flags.
BASS_SAMPLE_FLOAT	Use 32-bit floating-point sample data. Floating-point channels 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 BASS_Init , and the stream must be mono. The SPEAKER flags cannot be used together with this 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 DirectX 8 effects. See the DX8 effect implementations section for details. Use BASS_ChannelSetFX 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 BASS_ChannelGetData 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 ignored.
BASS_SPEAKER_xxx	<u>Speaker assignment flags</u> . 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 <u>BASS_CONFIG_ASYNCFILE_BUFFER</u> config option. This flag only applies when the <u>BASS_CONFIG_WMA_BASSFILE</u> config option is enabled, and is ignored when streaming from memory (<i>mem = TRUE</i>).
BASS_UNICODE	<i>file</i> , <i>user</i> and <i>pass</i> are in UTF-16 format. Otherwise they are ANSI.
user	Username to use in connecting to the server... if either this or <i>pass</i> is NULL then no username/password is sent to the server.
pass	Password to use in connecting to the server.

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_WMA	The Windows Media modules (v9 or above) are not installed.
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	One or more of the parameters are invalid.
BASS_ERROR_FILEOPEN	The file could not be opened.
BASS_ERROR_FILEFORM	The file's format is not recognised/supported.
BASS_ERROR_CODEEC	There is no appropriate codec installed to decode the file. Try installing the latest Windows Media codecs.
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_WMA_DENIED	Access was denied. Check the <i>user</i> and <i>pass</i> .
BASS_ERROR_WMA_LICENSE	The WMA file cannot be played because it is protected.

BASS_ERROR_UNKNOWN

Some other mystery problem!

Remarks

This function is identical to [BASS_WMA_StreamCreateFile](#), but with the additional authentication options.

See also

[BASS_WMA_StreamCreateFile](#)

BASS_WMA_StreamCreateFileUser

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

```
HSTREAM BASS_WMA_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
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 cannot be used together with this 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 Dirac 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 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 in 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 cannot 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_WMA	The Windows Media modules (v9 or above) are not installed.
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	<i>system</i> is not valid.
BASS_ERROR_FILEFORM	The file's format is not recognised/supported.
BASS_ERROR_CODEEC	There is no appropriate codec installed to decode the file. Try installing the latest Windows Media codecs.
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_WMA_LICENSE	The WMA file can not be played because it is protected.
BASS_ERROR_UNKNOWN	Some other mystery problem!

See also

[BASS_WMA_StreamCreateFile](#)

[BASS_ChannelGetInfo](#), [BASS_ChannelGetLength](#), [BASS_ChannelGetTags](#),
[BASS_ChannelPlay](#), [BASS_ChannelSetAttribute](#), [BASS_ChannelSetDSP](#),
[BASS_ChannelSetFX](#), [BASS_ChannelSetLink](#), [BASS_StreamFree](#),
[BASS_StreamPutFileData](#), [BASS_FILEPROCS](#) structure,
[BASS_CONFIG_NET_BUFFER](#)

WMA syncs - BASS_ChannelSetSync

Syncs are set on WMA streams in exactly the same way as on any other stream, using [BASS_ChannelSetSync](#). The following is a list of the types of sync supported on WMA streams.

Sync types, with *param* and [SYNCPROC data](#) definitions.

BASS_SYNC_WMA_META

mixtime only

Sync when a *mid-stream* tag (script) is encountered in a WMA stream. The tag is available from [BASS_ChannelGetTags](#) (BASS_TAG_WMA_META).

param : not used. **data** : not used.

BASS_SYNC_WMA_CHANGE

mixtime only

Sync on a track change in a server-side playlist. Updated tags are available via [BASS_ChannelGetTags](#).

param : not used. **data** : not used.

The BASS_SYNC_POS, BASS_SYNC_END, BASS_SYNC_SLIDE, BASS_SYNC_STALL, BASS_SYNC_DOWNLOAD and BASS_SYNC_FREE sync types are also supported on WMA streams, as described in the [BASS_ChannelSetSync](#) documentation.

Plugin system

As well as providing dedicated stream creation functions, BASSWMA supports the BASS plugin system, adding WMA file support to the standard BASS stream and sample creation functions: [BASS_StreamCreateFile](#), [BASS_StreamCreateURL](#), [BASS_StreamCreateFileUser](#), and [BASS_SampleLoad](#). This is enabled using the [BASS_PluginLoad](#) function.

BASS_WMA_EncodeClose

Finishes encoding and closes the file or network port.

```
BOOL BASS_WMA_EncodeClose(  
    HWMENCODE handle  
);
```

Parameters

handle The encoder handle.

Return value

If successful, TRUE is returned, else FALSE is returned. Use [BASS_ErrorGetCode](#) to get the error code.

Error codes

BASS_ERROR_HANDLE *handle* is not valid.

See also

[BASS_WMA_EncodeOpen](#), [BASS_WMA_EncodeOpenFile](#),
[BASS_WMA_EncodeOpenNetwork](#)

BASS_WMA_EncodeGetClients

Retrieves the number of clients currently connected to the encoder.

```
DWORD BASS_WMA_EncodeGetClients(  
    HWMENCODE handle  
);
```

Parameters

handle The encoder handle.

Return value

If successful, the number of clients is returned, else -1 is returned. Use [BASS_ErrorGetCode](#) to get the error code.

Error codes

BASS_ERROR_HANDLE	<i>handle</i> is not valid.
BASS_ERROR_NOTAVAIL	The encoder was not created with <u>BASS_WMA_EncodeOpenNetwork</u> .
BASS_ERROR_UNKNOWN	Some other mystery problem!

See also

[BASS_WMA_EncodeOpenNetwork](#), [BASS_WMA_EncodeSetNotify](#)

BASS_WMA_EncodeGetPort

Retrieves the network port for clients to connect to.

```
DWORD BASS_WMA_EncodeGetPort(  
    HWMENCODE handle  
);
```

Parameters

handle The encoder handle.

Return value

If successful, the port number is returned, else 0 is returned. Use [BASS_ErrorGetCode](#) to get the error code.

Error codes

BASS_ERROR_HANDLE *handle* is not valid.

BASS_ERROR_NOTAVAIL The encoder is not a network encoder, so no port is being used.

See also

[BASS_WMA_EncodeOpenNetwork](#)

BASS_WMA_EncodeGetRates

Retrieves the WMA encoding bitrates available for a specified sample format.

```
DWORD *BASS_WMA_EncodeGetRates(  
    DWORD freq,  
    DWORD chans,  
    DWORD flags  
);
```

Parameters

<code>freq</code>	The sample rate, or a BASS channel handle if the <code>BASS_WMA_ENCODE_SOURCE</code> flag is specified.
<code>chans</code>	The number of channels.
<code>flags</code>	A combination of these flags.
<code>BASS_WMA_ENCODE_SOURCE</code>	Use the sample format of the BASS channel with the handle in <i>freq</i> . The <i>chans</i> parameter is ignored.
<code>BASS_WMA_ENCODE_STANDARD</code>	Get available bitrates for standard WMA encoding. If neither this or the <code>BASS_WMA_ENCODE_PRO</code> flag is specified, then the bitrates available for either codec are returned.
<code>BASS_WMA_ENCODE_PRO</code>	Get available bitrates for WMA Pro encoding.
<code>BASS_WMA_ENCODE_24BIT</code>	Get available bitrates for 24-bit encoding, else 16-bit encoding rates.
<code>BASS_WMA_ENCODE_RATES_VBR</code>	Get available VBR (Variable BitRate) quality settings, else CBR (Constant BitRate) rates

Return value

If successful, a pointer to an array of the available bitrates is returned (terminated by a 0), else NULL is returned. Use [BASS_ErrorGetCode](#) to get the error code.

Error codes

BASS_ERROR_WMA	The Windows Media modules (v9 or above) are not installed.
BASS_ERROR_NOTAVAIL	No codec could be found to support the specified sample format.
BASS_ERROR_UNKNOWN	Some other mystery problem!

Remarks

When requesting VBR rates, the rates returned are quality settings. For example, 10 = 10% quality, 25 = 25% quality, etc... 100% quality is lossless.

The WMA codec expects 16-bit or 24-bit sample data depending on the `BASS_WMA_ENCODE_24BIT` flag, but `BASSWMA` will accept 8-bit, 16-bit or floating-point data, and convert it to the appropriate format. Of course, it makes little sense to encode 8-bit or 16-bit data in 24-bit.

The WMA codec currently supports the following sample rates: 8000, 11025, 16000, 22050, 32000, 44100, 48000, 88200, 96000. And the following number of channels: 1, 2, 6, 8. But not all combinations of these are supported. To encode other sample formats, the data will first have to be resampled to a supported format.

WMA Pro gives better quality than the standard WMA codec. Support for multi-channel (more than stereo) and 24-bit encoding is also only available with WMA Pro.

Example

List the CBR bitrates available at 44100hz 16-bit stereo.

```
DWORD *rates=BASS_WMA_EncodeGetRates(44100, 2, 0); // get a pointer
if (rates)
    while (*rates) {
        printf("%d\n", *rates); // display the rate
        rates++; // move on to the next rate
    }
```

See also

[BASS_WMA_EncodeOpen](#), [BASS_WMA_EncodeOpenFile](#),
[BASS_WMA_EncodeOpenNetwork](#), [BASS_WMA_EncodeOpenPublish](#)

BASS_WMA_EncodeOpen

Initializes WMA encoding to a user defined function.

```
HWMENCODE BASS_WMA_EncodeOpen(  
    DWORD freq,  
    DWORD chans,  
    DWORD flags,  
    DWORD bitrate,  
    WMENCODEPROC *proc,  
    void *user  
);
```


Parameters

<code>freq</code>	The sample rate, or a BASS channel handle if the <code>BASS_WMA_ENCODE_SOURCE</code> flag is specified.
<code>chans</code>	The number of channels. More than stereo requires WMA Pro or PCM.
<code>flags</code>	A combination of these flags.
<code>BASS_SAMPLE_8BITS</code>	8-bit sample data. If neither the <code>BASS_SAMPLE_FLOAT</code> flag is specified, then 16-bit data is encoded.
<code>BASS_SAMPLE_FLOAT</code>	32-bit floating-point sample data.
<code>BASS_WMA_ENCODE_SOURCE</code>	Use the BASS channel with the handle in <code>freq</code> as the encoder's source. The <code>chans</code> parameter is ignored, as are the <code>BASS_SAMPLE_8BITS</code> and <code>BASS_SAMPLE_FLOAT</code> flags. If the BASSenc add-on is loaded, the BASS_CONFIG_ENCODE_PRIORITY setting is used to determine which channel's DSP chain the encoding is performed in, otherwise priority is used.
<code>BASS_WMA_ENCODE_STANDARD</code>	Use standard WMA encoding. If neither this or the <code>BASS_WMA_ENCODE_PRO</code> flag is specified, then either WMA Standard or WMA Pro could be used, depending on whether the encoder supports the requested sample rate and bitrate.
<code>BASS_WMA_ENCODE_PRO</code>	Use WMA Pro encoding.
<code>BASS_WMA_ENCODE_PCM</code>	Write uncompressed PCM data to the ASF container. <code>bitrate</code> is ignored, but it should be non-0.
<code>BASS_WMA_ENCODE_24BIT</code>	Encode in 24-bit, else 16-bit. 24-bit encoding requires WMA Pro or WMA Pro 96.

BASS_WMA_ENCODE_SCRIPT	Enable the specification of tag stream (after encoding has been completed).
BASS_WMA_ENCODE_QUEUE	Queue data to feed the encoder asynchronously. This prevents BASS_WMA_EncodeWrite from being blocked by the encoder, but the application should control the amount of data which is encoded, as it is not recommended to queue too much data for the encoder to handle.
bitrate	The encoding bitrate, or VBR quality (100 or less).
proc	The user defined function to receive the encoded data.
user	User instance data to pass to the callback function.

Return value

If successful, the new encoder's handle is returned, else FALSE is returned. Use [BASS_ErrorGetCode](#) to get the error code.

Error codes

BASS_ERROR_WMA	The Windows Media modules (v9 or above) are not installed.
BASS_ERROR_NOTAVAIL	No codec could be found to support the requested sample format and bitrate.
BASS_ERROR_UNKNOWN	Some other mystery problem!

Remarks

Encoding *to a user defined function* allows any storage or delivery method to be used for the encoded WMA data. For example, encoding to memory.

The WMA codec expects 16-bit or 24-bit sample data depending on the `BASS_WMA_ENCODE_24BIT` flag, but BASSWMA will accept 8-bit, 16-bit or floating-point data, and convert it to the appropriate format. Use [BASS_WMA_EncodeGetRates](#) to retrieve a list of the encoding bitrates available for a specific sample format.

Use [BASS_WMA_EncodeSetTag](#) to set tags, [BASS_WMA_EncodeWrite](#) to encode sample data, and [BASS_WMA_EncodeClose](#) to finish encoding.

See also

[BASS_WMA_EncodeClose](#), [BASS_WMA_EncodeGetRates](#),
[BASS_WMA_EncodeOpenFile](#), [BASS_WMA_EncodeOpenNetwork](#),
[BASS_WMA_EncodeSetTag](#), [BASS_WMA_EncodeWrite](#),
[WMENCODEPROC](#) callback

BASS_WMA_EncodeOpenFile

Initializes WMA encoding to a file.

```
HWMENCODE BASS_WMA_EncodeOpenFile(  
    DWORD freq,  
    DWORD chans,  
    DWORD flags,  
    DWORD bitrate,  
    char *file  
);
```

Parameters

<code>freq</code>	The sample rate, or a BASS channel handle if the <code>BASS_WMA_ENCODE_SOURCE</code> flag is specified.
<code>chans</code>	The number of channels. More than stereo requires WMA Pro or PCM.
<code>flags</code>	A combination of these flags.
<code>BASS_SAMPLE_8BITS</code>	8-bit sample data. If neither the <code>BASS_SAMPLE_FLOAT</code> flag is specified, then 16-bit data is encoded.
<code>BASS_SAMPLE_FLOAT</code>	32-bit floating-point sample data.
<code>BASS_WMA_ENCODE_SOURCE</code>	Use the BASS channel with the handle in <code>freq</code> as the encoder's source. The <code>chans</code> parameter is ignored, as are the <code>BASS_SAMPLE_8BITS</code> and <code>BASS_SAMPLE_FLOAT</code> flags. If the BASSenc add-on is loaded, the BASS_CONFIG_ENCODE_PRIORITY setting is used to determine which channel's DSP chain the encoding is performed in, otherwise priority is used.
<code>BASS_WMA_ENCODE_STANDARD</code>	Use standard WMA encoding. If neither this or the <code>BASS_WMA_ENCODE_PRO</code> flag is specified, then either WMA Standard or Pro could be used, depending on whether the encoder supports the requested sample rate and bitrate.
<code>BASS_WMA_ENCODE_PRO</code>	Use WMA Pro encoding.
<code>BASS_WMA_ENCODE_PCM</code>	Write uncompressed PCM data to the ASF container. <code>bitrate</code> is ignored, but it should be non-0.
<code>BASS_WMA_ENCODE_24BIT</code>	Encode in 24-bit, else 16-bit. 24-bit encoding requires WMA Pro or WMA Pro 96.

BASS_WMA_ENCODE_SCRIPT	Enable the specification of tag stream (after encoding has been completed).
BASS_WMA_ENCODE_QUEUE	Queue data to feed the encoder asynchronously. This prevents BASS_WMA_EncodeWrite from being blocked by the encoder, but the application should control the amount of data which data is encoded, as it is not recommended to queue too much data for the encoder to handle.
BASS_UNICODE	<i>file</i> is a Unicode (UTF-16) filename.
bitrate	The encoding bitrate, or VBR quality (100 or less).
file	The filename to write.

Return value

If successful, the new encoder's handle is returned, else FALSE is returned. Use [BASS_ErrorGetCode](#) to get the error code.

Error codes

BASS_ERROR_WMA	The Windows Media modules (v9 or above) are not installed.
BASS_ERROR_NOTAVAIL	No codec could be found to support the requested sample format and bitrate.
BASS_ERROR_CREATE	Could not create the file to write the WMA stream.
BASS_ERROR_UNKNOWN	Some other mystery problem!

Remarks

The WMA codec expects 16-bit or 24-bit sample data depending on the `BASS_WMA_ENCODE_24BIT` flag, but BASSWMA will accept 8-bit, 16-bit or floating-point data, and convert it to the appropriate format. Use [BASS_WMA_EncodeGetRates](#) to retrieve a list of the encoding bitrates available for a specific sample format.

Use [BASS_WMA_EncodeSetTag](#) to set tags, [BASS_WMA_EncodeWrite](#) to encode sample data, and [BASS_WMA_EncodeClose](#) to finish encoding and close the file.

Example

Initialize encoding 44100hz 16-bit stereo sample data at 128kb/s to a file called "blah.wma".

```
HWMENCODE encoder=BASS_WMA_EncodeOpenFile(44100, 2, 0, 128000, "bla
```

See also

[BASS_WMA_EncodeClose](#), [BASS_WMA_EncodeGetRates](#),
[BASS_WMA_EncodeOpen](#), [BASS_WMA_EncodeOpenNetwork](#),
[BASS_WMA_EncodeSetTag](#), [BASS_WMA_EncodeWrite](#)

BASS_WMA_EncodeOpenNetwork

Initializes WMA encoding to the network.

```
HWMENCODE BASS_WMA_EncodeOpenNetwork(  
    DWORD freq,  
    DWORD chans,  
    DWORD flags,  
    DWORD bitrate,  
    DWORD port,  
    DWORD clients  
);
```

Parameters

<code>freq</code>	The sample rate, or a BASS channel handle if the <code>BASS_WMA_ENCODE_SOURCE</code> flag is specified.
<code>chans</code>	The number of channels. More than stereo requires WMA Pro or PCM.
<code>flags</code>	A combination of these flags.
<code>BASS_SAMPLE_8BITS</code>	8-bit sample data. If neither the <code>BASS_SAMPLE_FLOAT</code> flag nor this flag is specified, then 16-bit data is encoded.
<code>BASS_SAMPLE_FLOAT</code>	32-bit floating-point sample data.
<code>BASS_WMA_ENCODE_SOURCE</code>	Use the BASS channel with the handle <code>hChannel</code> in <code>freq</code> as the encoder's source. If this flag is specified, the <code>chans</code> parameter is ignored, as are the <code>BASS_SAMPLE_8BITS</code> and <code>BASS_SAMPLE_FLOAT</code> flags. If the BASSenc add-on is loaded, the <code>BASS_CONFIG_ENCODE_PRIORITY</code> setting is used to determine which channel's DSP chain the encoding is performed in, otherwise priority encoding is used.
<code>BASS_WMA_ENCODE_STANDARD</code>	Use standard WMA encoding. If this or the <code>BASS_WMA_ENCODE_PRO</code> flag is specified, then either WMA Standard or WMA Pro could be used, depending on whether the encoder supports the requested sample rate and bitrate.
<code>BASS_WMA_ENCODE_PRO</code>	Use WMA Pro encoding.
<code>BASS_WMA_ENCODE_PCM</code>	Write uncompressed PCM data to the ASF container. <code>bitrate</code> is ignored; it should be non-0.
<code>BASS_WMA_ENCODE_24BIT</code>	Encode in 24-bit, else 16-bit. 24-bit encoding requires WMA Pro.

	<code>BASS_WMA_ENCODE_SCRIPT</code>	Enable the specification of tag stream (after encoding has be
<code>bitrate</code>		The encoding bitrate.
<code>port</code>		The port number for clients to connect to... 0 = let the system choose a p
<code>clients</code>		The maximum number of clients (up to 50) that can be connected.

Return value

If successful, the new encoder's handle is returned, else FALSE is returned. Use [BASS_ErrorGetCode](#) to get the error code.

Error codes

BASS_ERROR_WMA	The Windows Media modules (v9 or above) are not installed.
BASS_ERROR_ILLPARAM	<i>clients</i> is invalid.
BASS_ERROR_NOTAVAIL	No codec could be found to support the requested sample format and bitrate.
BASS_ERROR_UNKNOWN	Some other mystery problem!

Remarks

If you chose to let the system select a port, you can retrieve the port number using [BASS_WMA_EncodeGetPort](#).

The WMA codec expects 16-bit or 24-bit sample data depending on the `BASS_WMA_ENCODE_24BIT` flag, but BASSWMA will accept 8-bit, 16-bit or floating-point data, and convert it to the appropriate format. Use [BASS_WMA_EncodeGetRates](#) to retrieve a list of the encoding bitrates available for a specific sample format. VBR encoding is not recommended for network encoding.

Use [BASS_WMA_EncodeSetTag](#) to set tags, [BASS_WMA_EncodeWrite](#) to encode sample data, and [BASS_WMA_EncodeClose](#) to finish encoding and close the network port.

The `BASS_WMA_ENCODE_QUEUE` flag is not necessary with this function as the data is always queued and fed to the encoder asynchronously.

Example

Initialize encoding 44100hz 16-bit stereo sample data at 128kb/s, using a system-chosen port, and allowing up to 5 clients.

```
HWMENCODE encoder=BASS_WMA_EncodeOpenNetwork(44100, 2, 0, 128000, 0
```

See also

[BASS_WMA_EncodeClose](#), [BASS_WMA_EncodeGetClients](#),
[BASS_WMA_EncodeGetPort](#), [BASS_WMA_EncodeGetRates](#),
[BASS_WMA_EncodeOpen](#), [BASS_WMA_EncodeOpenFile](#),
[BASS_WMA_EncodeOpenNetworkMulti](#), [BASS_WMA_EncodeOpenPublish](#),
[BASS_WMA_EncodeSetTag](#), [BASS_WMA_EncodeWrite](#)

BASS_WMA_EncodeOpenNetworkMulti

Initializes WMA encoding to the network, using multiple bitrates.

```
HWMENCODE BASS_WMA_EncodeOpenNetworkMulti(  
    DWORD freq,  
    DWORD chans,  
    DWORD flags,  
    DWORD *bitrates,  
    DWORD port,  
    DWORD clients  
);
```

Parameters

<code>freq</code>	The sample rate, or a BASS channel handle if the <code>BASS_WMA_ENCODE_SOURCE</code> flag is specified.
<code>chans</code>	The number of channels. More than stereo requires WMA Pro or PCM.
<code>flags</code>	A combination of these flags.
<code>BASS_SAMPLE_8BITS</code>	8-bit sample data. If neither <code>BASS_SAMPLE_FLOAT</code> nor <code>BASS_SAMPLE_8BITS</code> is specified, then 16-bit data is used.
<code>BASS_SAMPLE_FLOAT</code>	32-bit floating-point sample data.
<code>BASS_WMA_ENCODE_SOURCE</code>	Use the BASS channel with <code>handle</code> in <code>freq</code> as the encoder's source. If <code>BASS_WMA_ENCODE_SOURCE</code> is specified, <code>chans</code> parameter is ignored, and <code>BASS_SAMPLE_8BITS</code> and <code>BASS_SAMPLE_FLOAT</code> flags are ignored. If the BASSenc add-on is loaded, the BASS_CONFIG_ENCODE_PRIORITY setting is used to determine which channel's DSP chain the encoding is performed, otherwise priority is used.
<code>BASS_WMA_ENCODE_STANDARD</code>	Use standard WMA encoding. If <code>BASS_WMA_ENCODE_STANDARD</code> or <code>BASS_WMA_ENCODE_PRO</code> is specified, then either WMA Standard or WMA Pro could be used, depending on whether the encoder supports the requested sample rate and bitrate.
<code>BASS_WMA_ENCODE_PRO</code>	Use WMA Pro encoding.
<code>BASS_WMA_ENCODE_PCM</code>	Write uncompressed PCM data to the ASF container. The <code>bitrates</code> parameter is ignored except that it should be set to 0.
<code>BASS_WMA_ENCODE_24BIT</code>	Encode in 24-bit, else 16-bit. 24-bit encoding requires WMA Pro.

	BASS_WMA_ENCODE_SCRIPT	Enable the specification of ta stream (after encoding has be
bitrates	Array of encoding bitrates to use, terminated with a 0.	
port	The port number for clients to connect to... 0 = let the system choose a p	
clients	The maximum number of clients (up to 50) that can be connected.	

Return value

If successful, the new encoder's handle is returned, else FALSE is returned. Use [BASS_ErrorGetCode](#) to get the error code.

Error codes

BASS_ERROR_WMA	The Windows Media modules (v9 or above) are not installed.
BASS_ERROR_ILLPARAM	<i>clients</i> is invalid.
BASS_ERROR_NOTAVAIL	No codec could be found to support the requested sample format and bitrate.
BASS_ERROR_UNKNOWN	Some other mystery problem!

Remarks

This function is identical to [BASS_WMA_EncodeOpenNetwork](#), but with the additional ability to specify multiple bitrates.

When encoding/broadcasting in multiple bitrates, the user will automatically get the best available bitrate for their bandwidth.

Example

Initialize encoding 44100hz 16-bit stereo sample data at 128kb/s and 64kb/s, using a system-chosen port, and allowing up to 5 clients.

```
DWORD bitrates[3]={128000,64000,0}; // the bitrates  
HWMENCODE encoder=BASS_WMA_EncodeOpenNetworkMulti(44100, 2, 0, bitr
```

See also

[BASS_WMA_EncodeOpenNetwork](#)

BASS_WMA_EncodeOpenPublish

Initializes WMA encoding to a publishing point on a Windows Media server.

```
HWMENCODE BASS_WMA_EncodeOpenPublish(  
    DWORD freq,  
    DWORD chans,  
    DWORD flags,  
    DWORD bitrate,  
    char *url,  
    char *user,  
    char *pass  
);
```

Parameters

<code>freq</code>	The sample rate, or a BASS channel handle if the <code>BASS_WMA_ENCODE_SOURCE</code> flag is specified.
<code>chans</code>	The number of channels. More than stereo requires WMA Pro or PCM.
<code>flags</code>	A combination of these flags.
<code>BASS_SAMPLE_8BITS</code>	8-bit sample data. If neither the <code>BASS_SAMPLE_FLOAT</code> flag is specified, then 16-bit data is encoded.
<code>BASS_SAMPLE_FLOAT</code>	32-bit floating-point sample data.
<code>BASS_WMA_ENCODE_SOURCE</code>	Use the BASS channel with the handle in <code>freq</code> as the encoder's source. The <code>chans</code> parameter is ignored, as are the <code>BASS_SAMPLE_8BITS</code> and <code>BASS_SAMPLE_FLOAT</code> flags. If the BASSenc add-on is loaded, the BASS_CONFIG_ENCODE_PRIORITY setting is used to determine which channel's DSP chain the encoding is performed in, otherwise priority is used.
<code>BASS_WMA_ENCODE_STANDARD</code>	Use standard WMA encoding. If this or the <code>BASS_WMA_ENCODE_PRO</code> flag is specified, then either WMA Standard or WMA Pro could be used, depending on whether the encoder supports the requested sample rate and bitrate.
<code>BASS_WMA_ENCODE_PRO</code>	Use WMA Pro encoding.
<code>BASS_WMA_ENCODE_PCM</code>	Write uncompressed PCM data to the ASF container. <code>bitrate</code> is ignored, but it should be non-0.
<code>BASS_WMA_ENCODE_24BIT</code>	Encode in 24-bit, else 16-bit. 24-bit encoding requires WMA Pro or WMA Pro 96.

BASS_WMA_ENCODE_SCRIPT	Enable the specification of tag stream (after encoding has been completed).
BASS_UNICODE	<i>url</i> , <i>user</i> and <i>pass</i> are Unicode (16).
bitrate	The encoding bitrate.
url	URL of the publishing point on the Windows Media server.
user	Username to use in connecting to the server... if either this or <i>pass</i> is NULL, no username/password is sent to the server.
pass	Password to use in connecting to the server.

Return value

If successful, the new encoder's handle is returned, else FALSE is returned. Use [BASS_ErrorGetCode](#) to get the error code.

Error codes

BASS_ERROR_WMA	The Windows Media modules (v9 or above) are not installed.
BASS_ERROR_NOTAVAIL	No codec could be found to support the requested sample format and bitrate.
BASS_ERROR_FILEOPEN	Could not connect to the server.
BASS_ERROR_WMA_DENIED	Access was denied. Check the <i>user</i> and <i>pass</i> .
BASS_ERROR_WMA_PUBINIT	The server connection was not initialized properly. This can happen when connecting to the same server multiple times in quick succession. Try again after waiting a couple of seconds.
BASS_ERROR_UNKNOWN	Some other mystery problem!

Remarks

The WMA codec expects 16-bit or 24-bit sample data depending on the `BASS_WMA_ENCODE_24BIT` flag, but BASSWMA will accept 8-bit, 16-bit or floating-point data, and convert it to the appropriate format. Use [BASS_WMA_EncodeGetRates](#) to retrieve a list of the encoding bitrates available for a specific sample format. VBR encoding is not recommended for network encoding.

Use [BASS_WMA_EncodeSetTag](#) to set tags, [BASS_WMA_EncodeWrite](#) to encode sample data, and [BASS_WMA_EncodeClose](#) to finish encoding and close the connection to the server.

The `BASS_WMA_ENCODE_QUEUE` flag is not necessary with this function as the data is always queued and fed to the encoder asynchronously.

See also

[BASS_WMA_EncodeClose](#), [BASS_WMA_EncodeGetRates](#),
[BASS_WMA_EncodeOpen](#), [BASS_WMA_EncodeOpenFile](#),
[BASS_WMA_EncodeOpenNetwork](#), [BASS_WMA_EncodeOpenPublishMulti](#),
[BASS_WMA_EncodeSetTag](#), [BASS_WMA_EncodeWrite](#)

BASS_WMA_EncodeOpenPublishMulti

Initializes WMA encoding to a publishing point on a Windows Media server, using multiple bitrates.

```
HWMENCODE BASS_WMA_EncodeOpenPublishMulti(  
    DWORD freq,  
    DWORD chans,  
    DWORD flags,  
    DWORD *bitrates,  
    char *url,  
    char *user,  
    char *pass  
);
```

Parameters

<code>freq</code>	The sample rate, or a BASS channel handle if the <code>BASS_WMA_ENCODE_SOURCE</code> flag is specified.
<code>chans</code>	The number of channels. More than stereo requires WMA Pro or PCM.
<code>flags</code>	A combination of these flags.
<code>BASS_SAMPLE_8BITS</code>	8-bit sample data. If neither <code>BASS_SAMPLE_8BITS</code> nor <code>BASS_SAMPLE_FLOAT</code> is specified, then 16-bit data is used.
<code>BASS_SAMPLE_FLOAT</code>	32-bit floating-point sample data.
<code>BASS_WMA_ENCODE_SOURCE</code>	Use the BASS channel with <code>handle</code> in <code>freq</code> as the encoder's source. If <code>chans</code> parameter is ignored, and <code>BASS_SAMPLE_8BITS</code> and <code>BASS_SAMPLE_FLOAT</code> flags are not specified, the BASSenc add-on is loaded, then the BASS_CONFIG_ENCODE setting is used to determine which channel's DSP chain the encoding is performed, otherwise priority channel is used.
<code>BASS_WMA_ENCODE_STANDARD</code>	Use standard WMA encoding. If <code>BASS_WMA_ENCODE_PRO</code> or <code>BASS_WMA_ENCODE_PCM</code> is specified, then either WMA Pro or PCM could be used, depending on whether the encoder supports the requested sample rate and bitrate.
<code>BASS_WMA_ENCODE_PRO</code>	Use WMA Pro encoding.
<code>BASS_WMA_ENCODE_PCM</code>	Write uncompressed PCM data to the ASF container. The <code>bitrates</code> parameter is ignored except that it should be set to 0.
<code>BASS_WMA_ENCODE_24BIT</code>	Encode in 24-bit, else 16-bit. 24-bit encoding requires WMA Pro.

BASS_WMA_ENCODE_SCRIPT	Enable the specification of ta stream (after encoding has be
BASS_UNICODE	<i>url</i> , <i>user</i> and <i>pass</i> are Unicoc 16).
bitrates	Array of encoding bitrates to use, terminated with a 0.
url	URL of the publishing point on the Windows Media server.
user	Username to use in connecting to the server... if either this or <i>pass</i> is NU no username/password is sent to the server.
pass	Password to use in connecting to the server.

Return value

If successful, the new encoder's handle is returned, else FALSE is returned. Use [BASS_ErrorGetCode](#) to get the error code.

Error codes

BASS_ERROR_WMA	The Windows Media modules (v9 or above) are not installed.
BASS_ERROR_NOTAVAIL	No codec could be found to support the requested sample format and bitrate.
BASS_ERROR_FILEOPEN	Could not connect to the server.
BASS_ERROR_WMA_DENIED	Access was denied. Check the <i>user</i> and <i>pass</i> .
BASS_ERROR_WMA_PUBINIT	The server connection was not initialized properly. This can happen when connecting to the same server multiple times in quick succession. Try again after waiting a couple of seconds.
BASS_ERROR_UNKNOWN	Some other mystery problem!

Remarks

This function is identical to [BASS_WMA_EncodeOpenPublish](#), but with the additional ability to specify multiple bitrates.

When encoding/broadcasting in multiple bitrates, the user will automatically get the best available bitrate for their bandwidth.

See also

[BASS_WMA_EncodeOpenPublish](#)

BASS_WMA_EncodeSetNotify

Sets a client connection notification callback on a network encoder.

```
BOOL BASS_WMA_EncodeSetNotify(  
    HWMENCODE handle,  
    CLIENTCONNECTPROC *proc,  
    void *user  
);
```

Parameters

- handle The encoder handle.
- proc User defined notification function... NULL = disable notifications.
- user User instance data to pass to the callback function.

Return value

If successful, TRUE is returned, else FALSE is returned. Use [BASS_ErrorGetCode](#) to get the error code.

Error codes

BASS_ERROR_HANDLE *handle* is not valid.

BASS_ERROR_NOTAVAIL The encoder is not a network encoder, so does not have clients.

Remarks

A previously set notification callback can be changed (or removed) at any time, by calling this function again.

See also

[BASS_WMA_EncodeGetClients](#), [CLIENTCONNECTPROC](#) callback

BASS_WMA_EncodeSetTag

Sets a tag in a WMA encoding.

```
BOOL BASS_WMA_EncodeSetTag(  
    HWMENCODE handle,  
    char *tag,  
    char *value,  
    DWORD form  
);
```

Parameters

handle	The encoder handle.	
tag	The tag to set. The standard WMA header tags are as follows.	
	Title	Content title.
	Author	Name of the content author.
	Description	Description of the content.
	Rating	Content rating.
	Copyright	Content copyright message.
	WM/AlbumTitle	Album title.
	WM/PromotionURL	URL to an HTML page containing related information.
	WM/AlbumCoverURL	URL to an HTML page containing an image of the album cover.
	WM/Genre	Genre of the music.
	WM/Year	Year of publication of the music.
value	The tag's text/data.	
form	The format of the <i>tag</i> and <i>value</i> strings.	
	BASS_WMA_TAG_ANSI	ANSI strings.
	BASS_WMA_TAG_UNICODE	UTF-16 strings.
	BASS_WMA_TAG_UTF8	UTF-8 strings.
	BASS_WMA_TAG_BINARY	<i>value</i> contains binary data rather than text. The length of the data is in the HIWORD.

Return value

If successful, TRUE is returned, else FALSE is returned. Use [BASS_ErrorGetCode](#) to get the error code.

Error codes

BASS_ERROR_HANDLE	<i>handle</i> is not valid.
BASS_ERROR_NOTAVAIL	The encoder does not have mid-stream tags enabled, so tags cannot be set once encoding has begun.
BASS_ERROR_ILLPARAM	<i>tag</i> and/or <i>value</i> is invalid.
BASS_ERROR_UNKNOWN	Some other mystery problem!

Remarks

Where the tags are located in the encoded stream depends on when this function is used. Calling this function before beginning encoding data puts the tags in the stream's header. Calling this function after encoding has begun puts the tags in the actual stream data, at the current encoding position.

Header tags must be set before encoding any data; no more header tags can be set once [BASS_WMA_EncodeWrite](#) has been called.

To set tags mid-stream (after encoding has begun), the `BASS_WMA_ENCODE_SCRIPT` flag needs to have been specified in the encoder's creation. A *mid-stream* tag typically used is "Caption", which get's displayed in Windows Media Player 9 and above (if the user has enabled captions).

When using a network encoder, it should be noted that while all header tags are sent to newly connecting clients, prior *mid-stream* tags are not. So if, for example, you're using the "Caption" tag to indicate the current song title, it should be sent at fairly regular intervals (not only at the start of the song).

On the playback side, *mid-stream* tags can be processed using [BASS_ChannelSetSync](#) (`BASS_SYNC_META`).

Example

Initialize encoding 44100hz 16-bit stereo sample data at 128kb/s to a file called "blah.wma", and set the title to "Blah".

```
HWMENCODE encoder=BASS_WMA_EncodeOpenFile(44100, 2, 0, 128000, "bla  
BASS_WMA_EncodeSetTag(encoder, "Title", "Blah", BASS_WMA_TAG_ANSI);
```


See also

[BASS_WMA_EncodeOpen](#), [BASS_WMA_EncodeOpenFile](#),
[BASS_WMA_EncodeOpenNetwork](#)

BASS_WMA_EncodeWrite

Encodes sample data, and writes it to the file or network.

```
BOOL BASS_WMA_EncodeWrite(  
    HWMENCODE handle,  
    void *buffer,  
    DWORD length  
);
```

Parameters

- handle The encoder handle.
- buffer The buffer containing the sample data.
- length The number of bytes in the buffer.

Return value

If successful, TRUE is returned, else FALSE is returned. Use [BASS_ErrorGetCode](#) to get the error code.

Error codes

BASS_ERROR_HANDLE	<i>handle</i> is not valid.
BASS_ERROR_MEM	There is insufficient memory.
BASS_ERROR_UNKNOWN	Some other mystery problem!

Remarks

There is generally no need to call this function if the `BASS_WMA_ENCODE_SOURCE` flag has been set on the encoder, as the encoder will automatically be fed the data that its source BASS channel produces.

See also

[BASS_WMA_EncodeOpen](#), [BASS_WMA_EncodeOpenFile](#),
[BASS_WMA_EncodeOpenNetwork](#)

CLIENTCONNECTPROC callback

User defined client connection notification callback function.

```
void CALLBACK ClientConnectProc(  
    HWMENCODE handle,  
    BOOL connect,  
    const char *ip  
    void *user  
);
```


Parameters

- handle The encoder handle.
- connect The client is connecting?
- ip The client's IP address... "xxx.xxx.xxx.xxx:port".
- user The user instance data given when [BASS_WMA_EncodeSetNotify](#) was called.

Remarks

A client connection notification can be used to keep track of who's connected, where they're from, and for long they've been connected.

Example

A callback function to log connections and disconnections.

```
void CALLBACK MyClientConnectProc(HWMENCODE handle, BOOL connect, c
    if (connect)
        printf("%s connected\n", ip);
    else
        printf("%s disconnected\n", ip);
}
...
BASS_WMA_EncodeSetNotify(handle, &MyClientConnectProc, 0); // set
```

See also

[BASS_WMA_EncodeSetNotify](#)

WMENCODEPROC callback

Encoded data processing callback function.

```
void CALLBACK WMEncodeProc(  
    HWMENCODE handle,  
    DWORD type,  
    const void *buffer,  
    DWORD length,  
    void *user  
);
```

Parameters

handle	The encoder handle.
type	The type of data to process, one of the following. BASS_WMA_ENCODE_HEAD The data is the header. BASS_WMA_ENCODE_DATA The data is encoded sample data. BASS_WMA_ENCODE_DONE The encoding has finished... <i>buffer</i> and <i>length</i> will both be 0.
buffer	A pointer to the data to process.
length	The length of the data.
user	The user instance data given when BASS_WMA_EncodeOpen was called.

Remarks

When encoding begins, an initial header is given. When encoding is completed, an updated header is given (with the duration info, etc). When encoding to a file (whether that is on disk or not), the initial header should be replaced by the updated one.

Example

A callback function to encode to a file.

```
void CALLBACK MyWMEncodeProc(HWMENCODE handle, DWORD type, const void* buffer,
{
    if (type==BASS_WMA_ENCODE_HEAD) {
        fseek(user, 0, SEEK_SET); // rewind to start of file
        fwrite(buffer, length, 1, user); // write the header
    } else if (type==BASS_WMA_ENCODE_DATA)
        fwrite(buffer, length, 1, user); // write encoded data
    else if (type==BASS_WMA_ENCODE_DONE)
        fclose(user); // done encoding - close the file
}
...
FILE *file=fopen("a_file.wma", "wb"); // created the file
BASS_WMA_EncodeOpen(44100, 2, 0, 64000, &MyWMEncodeProc, file); //
```

NOTE: This is just an example. It is simpler to use [BASS_WMA_EncodeOpenFile](#) to encode to a file.

See also

[BASS_WMA_EncodeOpen](#)

BASS_WMA_GetTags

Retrieves the tags from a WMA file.

```
char *BASS_WMA_GetTags(  
    char *file,  
    DWORD flags  
);
```

Parameters

`file` The filename.

`flags` A combination of these flags.

`BASS_TAG_WMA_CODEC` Get codec information rather than tags.

`BASS_UNICODE` *file* is a Unicode (UTF-16) filename.

Return value

If successful, the tags are returned, else NULL is returned. Use [BASS_ErrorGetCode](#) to get the error code.

Error codes

BASS_ERROR_WMA	The Windows Media modules (v9 or above) are not installed.
BASS_ERROR_FILEOPEN	The file could not be opened, or it is not a WMA file.
BASS_ERROR_UNKNOWN	Some other mystery problem!

Remarks

This function gives the same tags as [BASS_ChannelGetTags](#) (with BASS_TAG_WMA or BASS_TAG_WMA_CODEC), which is a pointer to a series of null-terminated UTF-8 strings, the final string ending with a double null. Unlike [BASS_ChannelGetTags](#), this function can also be used with DRM-protected WMA files without a DRM licence, as it does not require a stream to be created.

The memory used for the tags is reused by all calls of this function, so if the tags need to be retained across multiple calls, a copy should be made. It also means that this function is not thread-safe, that is it should not be called simultaneously from multiple threads.

Example

List a WMA file's tags.

```
char *tags=BASS_WMA_GetTags("a.wma", 0); // get the tags
if (tags)
    while (*tags) {
        printf("%s\n",tags); // display the tag
        tags+=strlen(tags)+1; // move on to next tag
    }
```

See also

[BASS_ChannelGetTags](#)