

DAW Host Information in RackAFX v6.8+

Will Pirkle

In RackAFX v6.8 and higher, you may now access information from the host DAW about the current audio processing and other audio and musical settings. This information is updated on each audio processing loop. VST3, AU and AAX all have separate sets of host information that is available. RackAFX also updates this kind of information for you, however it only passes the information that is common to all APIs. When using the Make VST, Make AU or Make AAX operations, you can custom tailor the information your plugin receives based on the plugin API you are implementing. You will note that some APIs contain more or less information than others (detailed below).

The host data that is common across all APIs is:

- tempo (beats per minute or BPM)
- time signature numerator (6 for 6/8 time)
- time signature denominator (8 for 6/8 time)
- absolute frame buffer index: the index value of the current audio frame of the current audio buffer that is being processed; if using *processAudioFrame()*, this value is updated on each sample interval, if using either of the buffer processing functions, this value is updated on each audio buffer (i.e. you need to keep track of your own sample-interval-specific values) — for example, if the audio driver is delivering 1024 samples per channel per buffer, the sequence would be 0, 1024, 2048, 4096...
- absolute frame buffer time: this is the time (in seconds) of the current absolute buffer index

The other APIs have specific information somewhat unique to them, so if you use the additional information from those APIs, you will need to understand what is available (detailed below) and some of the pitfalls and idiosyncrasies of each.

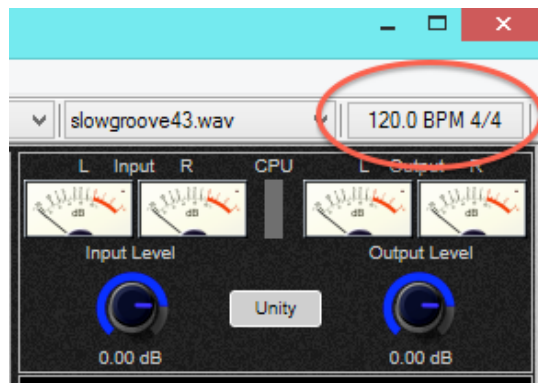
Host Information Structure: HOST_INFO

The host information is written into a structure at the start of each new audio buffer that is to be processed in your plugin. Your plugin includes a statically declared member variable that you can access to get this information. The datatype is `HOST_INFO` and the member variable is declared on your *CPlugin* base class as:

```
// --- info from host about current processing
HOST_INFO m_HostProcessInfo;
```

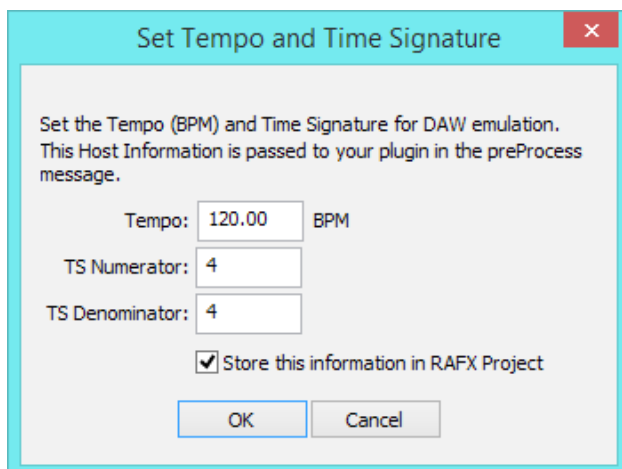
You can access this member variable at the start of each process cycle.

RackAFX, VST3, AU, AAX



In RackAFX you can set the tempo and time signature information by clicking the button in the upper right part of the UI as shown in Figure 1:

Figure 1: the tempo & time signature button in RackAFX v6.8



This brings up the dialog box shown in Figure 2 which is fairly straightforward. The checkbox “Store this information in RAFX Project” will write the data into your RackAFX project file so that it will always start up with these settings. You can change these values as audio is playing and they will be updated on your plugin, though this is really just “fake” information - RackAFX is not a DAW and does not actually implement any metronome or other musical timing events - this is just to allow you to experiment and debug your algorithms for use in the other APIs.

Figure 2: the tempo and time signature settings

Examining the HOST_INFO structure in pluginconstants.h, you can see where this information resides for you to use (it is heavily commented in the file, but most comments have been removed for this document for easier reading). This information is common to all APIs.

```
// --- common to all APIs
unsigned long long uAbsoluteFrameBufferIndex;
double dAbsoluteFrameBufferTime;
double dBPM;
float fTimeSigNumerator;
UINT uTimeSigDenomintor;
```

VST3

For your Make VST (and RAFX-as-VST-DLL) projects, there is additional information available for you. Make sure to read the VST documentation to understand which variables are always valid and which ones are optionally implemented in VST3 hosts. The variables are:

```
// --- VST3 Specific: note these use same variable names as VST3::struct
// ProcessContext
// see ..\VST3 SDK\pluginterfaces\vst\ivstprocesscontext.h
// for information on decoding these
//
unsigned int state;
long long systemTime;
double continousTimeSamples;
double projectTimeMusic;
double barPositionMusic;
double cycleStartMusic;
double cycleEndMusic;
unsigned int samplesToNextClock;
```

AU

For your Make AU projects, there is additional information available for you. Make sure to read the AU documentation for more information. In addition you may want to google different AU hosts to understand that there may be “interesting” behavior — for example, in Logic 9 and Logic Pro X, the flag that tells you whether audio is playing or not is “sticky” - once set to true, it remains true forever.

```
// --- AU Specific
//     see AUBase.h for definitions and information on decoding these
//
double dCurrentBeat;
bool bIsPlayingAU; // --- notoriously incorrect in Logic
bool bTransportStateChanged; // --- only notifies a change
unsigned int nDeltaSampleOffsetToNextBeat;
double dCurrentMeasureDownBeat;
bool bIsCycling;
double dCycleStartBeat;
double dCycleEndBeat;
```

AAX

For your Make AAX projects, there is additional information available for you. Make sure to read the AAX documentation for more information.

```
// --- AAX Specific
//     see AAX_ITransport.h for definitions and information on decoding these
bool bIsPlayingAAX; // --- flag if playing
long long nTickPosition; // --- "Tick" = 1/960000 of a quarter note
bool bLooping; // --- looping flag
long long nLoopStartTick;
long long nLoopEndTick ;
```

In addition, there are two more AAX host functions that can be called to glean more information about the DAW configuration however these incur some CPU processing cycles and should be used sparingly, therefore they are NOT called in the Make AAX projects, however comment blocks are placed in the code so you can examine them and perhaps relocate them to non-audio processing threads. See the AAX_ITransport.h file for more information on them.

Usage

On each process cycle (the start of *processAudioFrame()* or *processRackAFXBuffer()* or *processVST-Buffer()*, the member variables will be updated with the current host information. You may read these values and modify your code as you like. The values will be overwritten on each process cycle. The best way to start using these variables is to set the temp and time signature values in RackAFX, then use the debugger to step through a few rounds of audio processing so you can see how the variables are updated.