

AWE Core™

Cross-Platform Audio Processing Engine

API Version 8.B

Audio Weaver 8 comes with a clean new API for the AWE Core, further simplifying the software integration model. The new API reflects the three types of interactions between the product’s application code and the AWE Core library:

Configuration

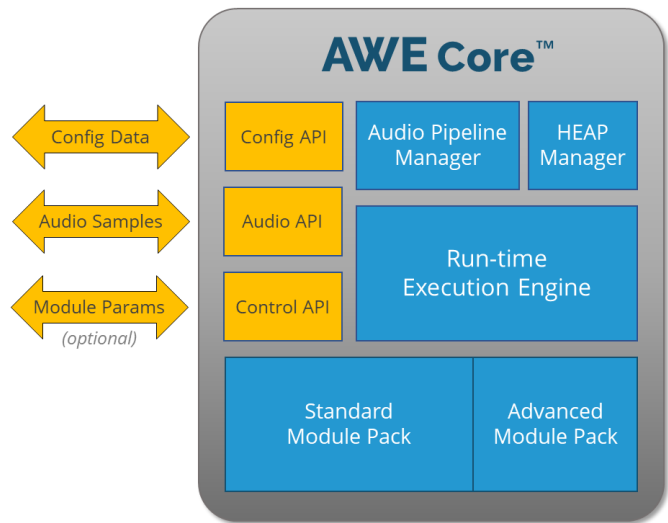
The AWE Core dynamically reconfigures its audio processing pipeline at runtime, based on configuration data (generated by AWE Designer™) that describes a user-created block-diagram

Audio I/O & Processing

Blocks of raw audio samples are imported, “pumped” through the active processing pipeline, and then exported back to the application as buffers of processed samples.

Module Control Data

The Control interface allows host applications to interact with module instances in the active processing pipeline, e.g. setting `mainMute.isMuted` to 1 when the Mute Button is pressed.



The following API summary is provided for high-level context. Please refer to the *AWE Core Integration Guide* and `AWECore.h` for full details.

```

Initialization
awe_init()
awe_initPin()
    
```

```

Module Control
awe_ctrlGetValue()
awe_ctrlSetValue()
awe_ctrlSetStatus()
awe_ctrlGetStatus()
awe_ctrlSetValueMask()
awe_ctrlGetValueMask()
awe_ctrlGetModuleClass()
awe_deferredSetCall()
    
```

```

Audio
awe_audioImportSamples()
awe_audioIsStarted()
awe_audioGetPumpMask()
awe_audioPump()
awe_audioExportSamples()
    
```

```

Configuration
awe_loadAWBfromArray()
awe_packetProcess()
awe_layoutGetChannelCount()
awe_layoutGetInputBlockSize()
awe_layoutGetInputSampleRate()
awe_layoutIsValid()
    
```