



Apple Pro Applications and Mac OS X Audio Unit Plug-ins Preliminary Developer Guidelines - Information subject to change

© 2003 Apple Computer, Inc.

This document provides guidelines for developers wishing to deploy their Audio Unit modules in the Final Cut Pro 4.0 application, the Soundtrack 1.0 application, or the Logic Audio 5.5 or 6.0 application.

For general information about the Mac OS X Core Audio architecture and Audio Unit plug-ins, please refer to the documentation and examples provided by Apple at <http://developer.apple.com/audio/>.

For the Final Cut Pro 4.0 application

For the Final Cut Pro 4.0 application, Audio Unit plug-in developers should be aware of these points:

1. The Final Cut Pro 4.0 application does not support any vendor-created custom user interface elements that are part of an Audio Unit plug-in. Instead, the user controls parameters for a plug-in with the standard user interface controls (sliders, checkboxes, and so on) within the Audio Filter Effects window. Audio Unit parameters that are not published for programmatic access have no corresponding control element appearing in the Audio Filter Effects window. Audio Units that are compatible with the Final Cut Pro 4.0 application appear in sub-folders labeled with the name of the manufacturer within the Audio Filter Effects window.
2. The Final Cut Pro 4.0 audio track and bussing architecture only supports monophonic channel processing of individual track items. This means that Audio Unit plug-ins, if they are to be used within the Final Cut Pro application, must provide monophonic channel input to monophonic channel output. Plug-ins that only implement stereo or other multi-channel processing are not loaded.
3. To support real-time adjustments during playback, an Audio Unit plug-in should implement the property 'kAudioUnitProperty_TailTime.' Audio playback and rendering in the Final Cut Pro application operate with an internal pre-roll audio computation method that ensures consistent sample values across edit points and during playback positioning. The application queries the plug-in for the value of this property to determine how many samples must be computed prior to the actual playback or edit position so that sample-accurate seaming is maintained.

Plug-ins that report 'not implemented' for this property load and operate subject to the following limits: if the application's audio playback quality preference is set to low-quality, you can adjust the parameters for the plug-in in real-time during playback, but sample-seaming artifacts may be audible; if the audio playback quality preference is set to medium or high in order to eliminate seaming artifacts, you must render items to which the plug-in is attached.

Units whose internal signal processing produces an infinite tail time, or whose internal state is indeterminate for a given sample offset, should report 'not implemented' for the 'kAudioUnitProperty_TailTime' property. These units cannot be seamed properly during playback and require item rendering in order to produce a consistent sample stream.

For more information on the 'kAudioUnitProperty_TailTime' property, see the Apple documentation on Mac OS X Core Audio architecture.

For the Soundtrack 1.0 application

For the Soundtrack 1.0 application, developers should note these points:

1. Soundtrack *does* allow for the use of custom Audio Unit user interface elements. In Soundtrack, this custom plug-in user interface is optional. It only appears when the user clicks the Advanced button in the generic Soundtrack user interface. (If the Audio Unit plug-in has no custom user interface elements, the Advanced button does not appear.) To ensure that an Audio Unit plug-in appears correctly in the generic Soundtrack user interface, developers should make sure that the parameter descriptions for the plug-in are complete and accurate.
2. Soundtrack requires that Audio Unit plug-ins have at least stereo input and stereo output channel capabilities. As long as an Audio Unit plug-in is capable of stereo input and stereo output, Soundtrack uses it correctly.
3. Soundtrack utilizes the Audio Unit property 'kAudioUnitProperty_TailTime' for a CPU usage-saving algorithm. An Audio Unit plug-in that does not implement this property still operates in Soundtrack. However, if the plug-in creates an audible tail after input ends (as in the case of reverb) and if the 'kAudioUnitProperty_TailTime' property is not implemented, Soundtrack may truncate the audio output.

For Logic Audio 5.5 or 6.0 application

For Logic Audio 5.5 or 6.0, these are some key points for developers:

1. The current reference hosts are Logic Platinum 5.5.1 or higher and Logic Platinum 6.0.1 or higher. Testing should be done with these versions.
2. Logic instantiates every Audio Unit once during startup but does not initialize it. To avoid long startup times, developers should put resource allocations into the initialization, not in the instantiation.
3. If available, Logic uses the custom Audio Unit interface. It also features a generic interface, that is used if no custom interface is available, or the user switches the interface view by menu selection. For the generic interface, the information from the parameter descriptions is used, so developers should provide a complete set of parameter information.

4. For plug-in parameter automation in combination with custom interfaces, it is essential that Audio Unit plug-ins send the events 'kAudioUnitCarbonViewEvent_MouseDownInControl' and 'kAudioUnitCarbonViewEvent_MouseUpInControl' and use the parameter listener scheme to inform the host of parameter changes.

5. Logic Audio currently supports mono-to-mono, mono-to-stereo and stereo-to-stereo Audio Units.

6. Logic uses the standard '.aupreset' file format for saving and loading presets. Developers should make sure that the property lists provided and accepted by the ClassInfo property are of the form described in the Audio Unit documentation.

7. Logic supports the latency and tail time features of the Audio Unit standard.

Document Revision v2.0 June 19, 2003