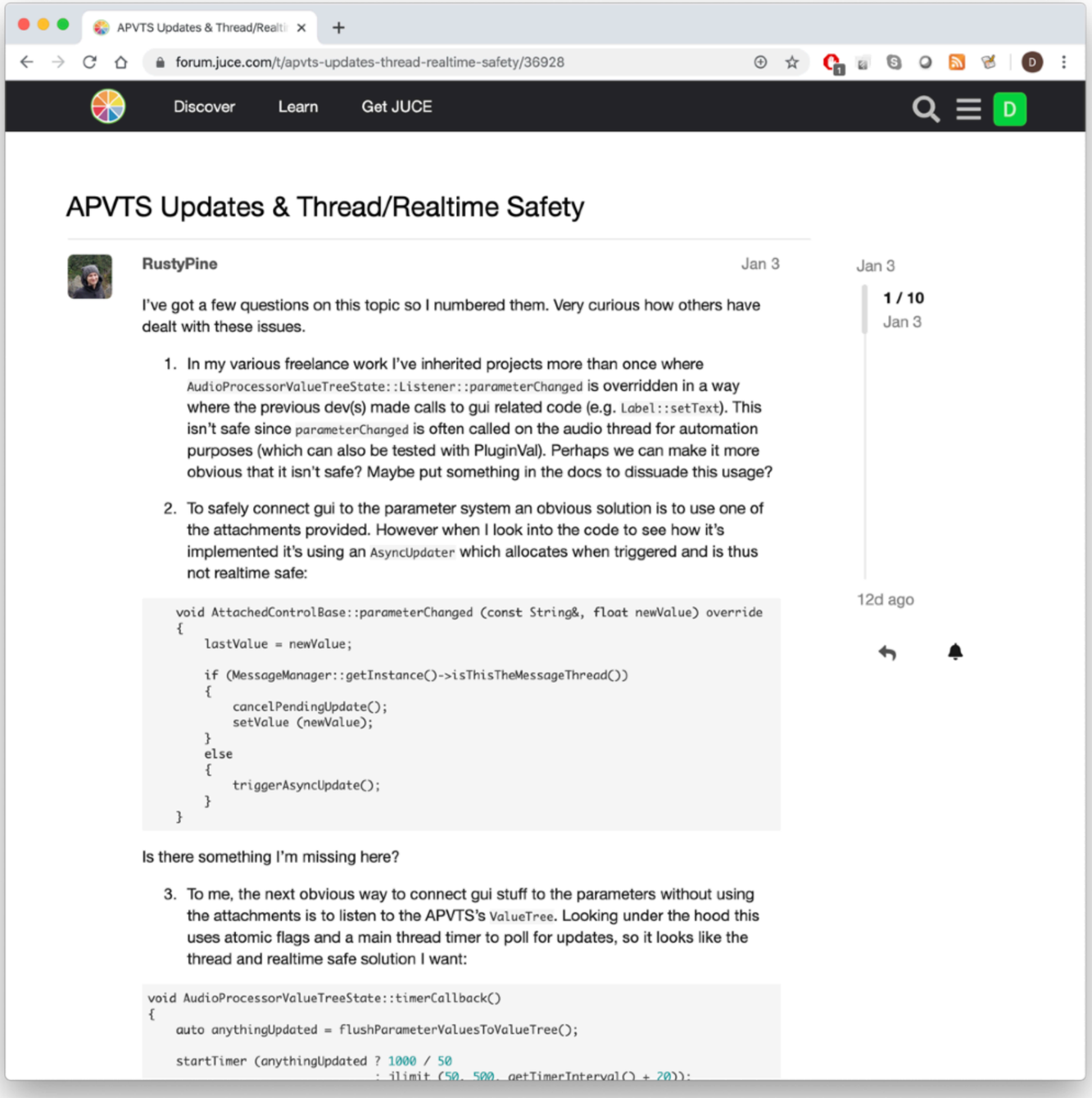
1. Background


APVTS Updates & Thread/Realti
forum.juce.com/t/apvts-updates-thread-realtime-safety/36928
Discover
Learn
Get JUCE
D
APVTS Updates & Thread/Realtime Safety
RustyPine
Jan 3
I've got a few questions on this topic so I numbered them. Very curious how others have dealt with these issues.
1. In my various freelance work I've inherited projects more than once where AudioProcessorValueTreeState::Listener::parameterChanged is overridden in a way where the previous dev(s) made calls to gui related code (e.g. Label::setText). This isn't safe since parameterChanged is often called on the audio thread for automation purposes (which can also be tested with PluginVal). Perhaps we can make it more obvious that it isn't safe? Maybe put something in the docs to dissuade this usage?
2. To safely connect gui to the parameter system an obvious solution is to use one of the attachments provided. However when I look into the code to see how it's implemented it's using an AsyncUpdater which allocates when triggered and is thus not realtime safe:
void AttachedControlBase::parameterChanged (const String&, float newValue) override
{
lastValue = newValue;
if (MessageManager::getInstance()->isThisTheMessageThread())
{
cancelPendingUpdate();
setValue (newValue);
}
else
{
triggerAsyncUpdate();
}
}
Is there something I'm missing here?
3. To me, the next obvious way to connect gui stuff to the parameters without using the attachments is to listen to the APVTS's ValueTree. Looking under the hood this uses atomic flags and a main thread timer to poll for updates, so it looks like the thread and realtime safe solution I want:
void AudioProcessorValueTreeState::timerCallback()
{
auto anythingUpdated = flushParameterValuesToValueTree();
startTimer (anythingUpdated ? 1000 / 50
Jan 3
1 / 10
Jan 3
12d ago