



ADC²³

HARRY MORLEY

BUILDING AN ACCESSIBLE JUCE APP

Get the repository (slides in docs/)






Repository URL: github.com/FocusriteGroup/accessible-juce-app-adc-23

Agenda

1. Demo of an app with default JUCE component accessibility
2. Dev tips for testing accessibility
3. Improving the app's accessibility
4. *Tour of an accessible app - Focusrite Control 2*
5. Q&A

About me

- Software Developer Focusrite, since 2019
 - Scarlett 3rd Gen
 - Vocaster
 - Scarlett 4th Gen 
- Musician 
 - Psych pop, rock, jazz, weird boops & digital mess 
- Love accessible software!

Demo

```
git checkout ':/Initial commit'
```

Testing accessibility

- OS-specific tools
 - macOS: Accessibility Inspector (Xcode Developer Tool)
developer.apple.com/xcode
 - Windows: Accessibility Insights
accessibilityinsights.io

Use a screen reader

...for a more representative experience.

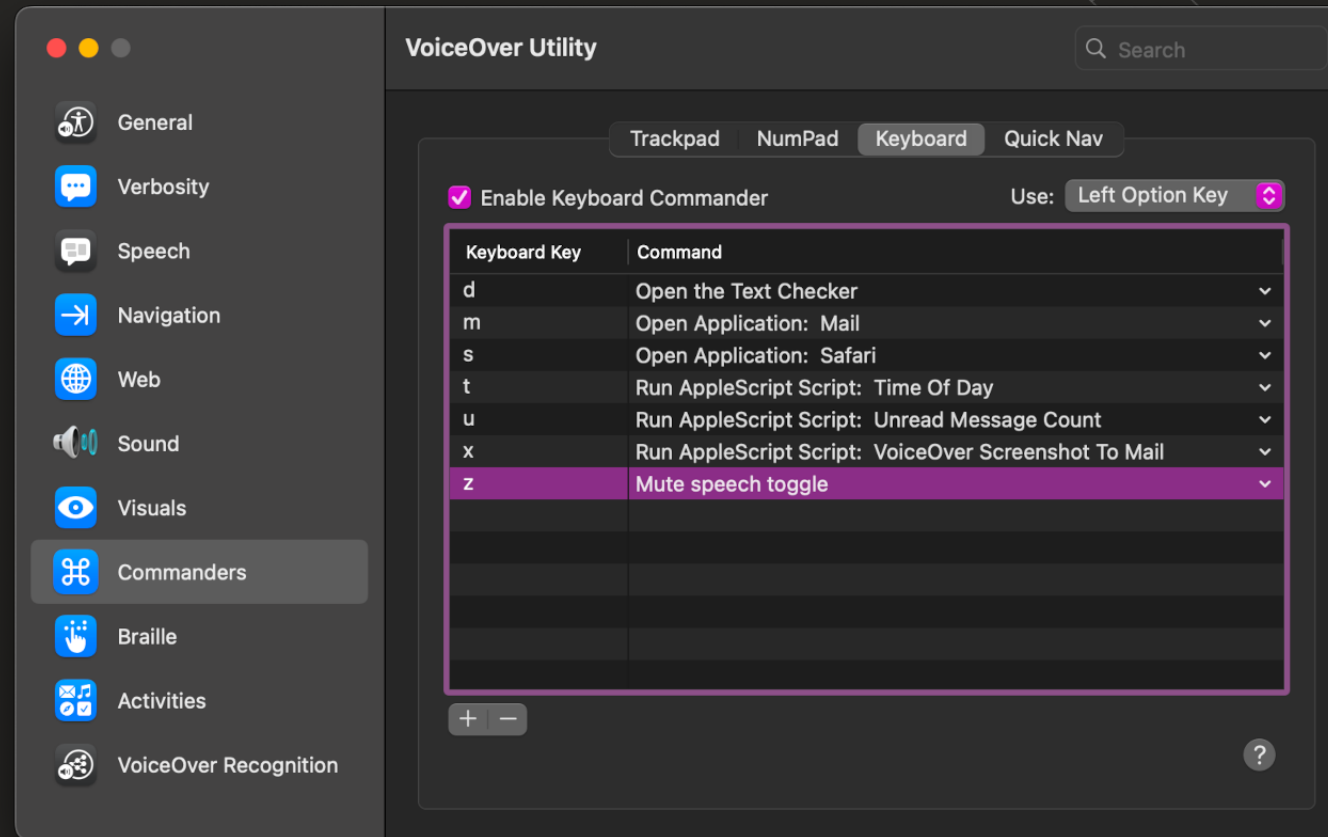
- macOS - VoiceOver
- Windows
 - Narrator
 - JAWS
 - NVDA

Interesting read: WebAIM Screen Reader Survey
webaim.org/projects/screenreadersurvey9/

Streamline your experience

- Do a tutorial (decent one often built-in to the screenreader)
- Learn the shortcuts, e.g. on macOS:
 - `cmd + F5` to toggle VoiceOver
 - VO key: `caps lock` or `ctrl + alt`
- Turn up the speed (if you wish)
- Know how to silence it (`ctrl` on mac)
- Set up toggle speech mute (*temporarily* use the caption panel instead)

Toggling speech mute



Back to the demo

Issues 🙄

- ✗ Can't interact with the keyboard alone
- ✗ Titles missing
- ✗ Unclear grouping ("mute" for what?)
- ✗ Not clear that buttons can be toggled
- ✗ (Windows) Expected the space bar would trigger buttons
- ✗ Sliders not receiving keyboard focus
- ✗ Strange focus order

Fixing the initial keyboard interaction

```
git checkout ':/Fix application window'
```

The (counterintuitive) fix

- Call `setWantsKeyboardFocus (false)` on the `DocumentWindow`
 - Otherwise keyboard focus becomes trapped on the window component
- Alternatively, inherit `activeWindowStatusChanged` and grab keyboard focus of the main content component when visible

Issues

- ✓ Can't interact with the keyboard alone - fixed!
- ✗ Titles missing
- ✗ Unclear grouping ("mute" for what?)
- ✗ Not clear that buttons can be toggled
- ✗ (Windows) Expected the space bar would trigger buttons
- ✗ Sliders not receiving keyboard focus
- ✗ Strange focus order

Adding titles

```
git checkout ':/Add titles'
```

Issues

- ✓ Can't interact with the keyboard alone - fixed!
- ✓ Titles missing - fixed!
- ✗ Unclear grouping ("mute" for what?)
- ✗ Not clear that buttons can be toggled
- ✗ (Windows) Expected the space bar would trigger buttons
- ✗ Sliders not receiving keyboard focus
- ✗ Strange focus order

Fixing the grouping

```
git checkout ':/Make channel strip a focus container'
```

Focus containers

- Components have the "group" role by default
- You have to specify the focus behaviour
 - `juce::Component::FocusContainerType::focusContainer`
 - `juce::Component::FocusContainerType::keyboardFocusContainer`

Side note: design considerations

- Windows screen readers generally keyboard focus-oriented (i.e. using tab to move around)
- Tab used for macOS/VoiceOver as well, but also VO + arrow keys, for traversing the layout

Issues

- ✓ Can't interact with the keyboard alone - fixed!
- ✓ Titles missing - fixed!
- ✓ Unclear grouping - fixed!
- ✗ Not clear that buttons can be toggled
- ✗ (Windows) Expected the space bar would trigger buttons
- ✗ Sliders not receiving keyboard focus
- ✗ Strange focus order

Fixing the buttons

```
git checkout ':/Support toggle button role'
```

Accessibility Handlers

- Write a custom accessibility handler from scratch

```
juce::AccessibilityHandler (juce::Component& componentToWrap,  
                             juce::AccessibilityRole accessibilityRole, // yaaas  
                             juce::AccessibilityActions actions = {},  
                             juce::Interfaces interfaces = {});
```

- Override `createAccessibilityHandler` in our custom button class:

```
std::unique_ptr<juce::AccessibilityHandler> createAccessibilityHandler () override;
```

- JUCE's Button accessibility handler is about 120 lines long...
 - We'd end up duplicating that

No need to reinvent the wheel

- In the cpp file of your custom button class...

```
#include <juce_gui_basics/detail/juce_ButtonAccessibilityHandler.h>

std::unique_ptr<juce::AccessibilityHandler> Button::createAccessibilityHandler () {
    return std::make_unique<juce::detail::ButtonAccessibilityHandler> (*this,
                                                                    isToggleable ()
                                                                    ? juce::AccessibilityRole::toggleButton
                                                                    : juce::AccessibilityRole::button);
}
```

- ⚠ This is using JUCE implementation detail and could change
 - Given it's a very specific class, maybe that's what we want
- ⚠ It could be removed in the future
 - Then we can just write our accessibility handler from scratch

Issues

- ✓ Can't interact with the keyboard alone - fixed!
- ✓ Titles missing - fixed!
- ✓ Unclear grouping - fixed!
- ✓ Not clear that buttons can be toggled - fixed!
- ✗ (Windows) Expected the space bar would trigger buttons
- ✗ Strange focus order
- ✗ Where are the sliders?

Making space bar trigger buttons

```
git checkout ':/Allow space bar'
```

Making space bar trigger buttons

- `juce::Button::addShortcut` doesn't do what you think!
- Instead, override `keyPressed` and make it trigger on `return` and `space`

```
bool keyPressed (const juce::KeyPress & key) override {
    if (! isEnabled ())
        return false;

    if (key == juce::KeyPress::returnKey || key == juce::KeyPress::spaceKey) {
        triggerClick ();
        return true;
    }

    return false;
}
```


Issues

- ✓ Can't interact with the keyboard alone - fixed!
- ✓ Titles missing - fixed!
- ✓ Unclear grouping - fixed!
- ✓ Not clear that buttons can be toggled - fixed!
- ✓ (Windows) Expected the space bar would trigger buttons - fixed!
- ✗ Sliders not receiving keyboard focus
- ✗ Strange focus order

Letting the sliders receive keyboard focus

```
git checkout ':/Allow sliders'
```

Issues

- ✓ Can't interact with the keyboard alone - fixed!
- ✓ Titles missing - fixed!
- ✓ Unclear grouping - fixed!
- ✓ Not clear that buttons can be toggled - fixed!
- ✓ (Windows) Expected the space bar would trigger buttons - fixed!
- ✓ Sliders not receiving keyboard focus - fixed!
- ✗ Strange focus order

Fixing the focus order

```
git checkout ':/Set explicit focus order'
```

Focus orders

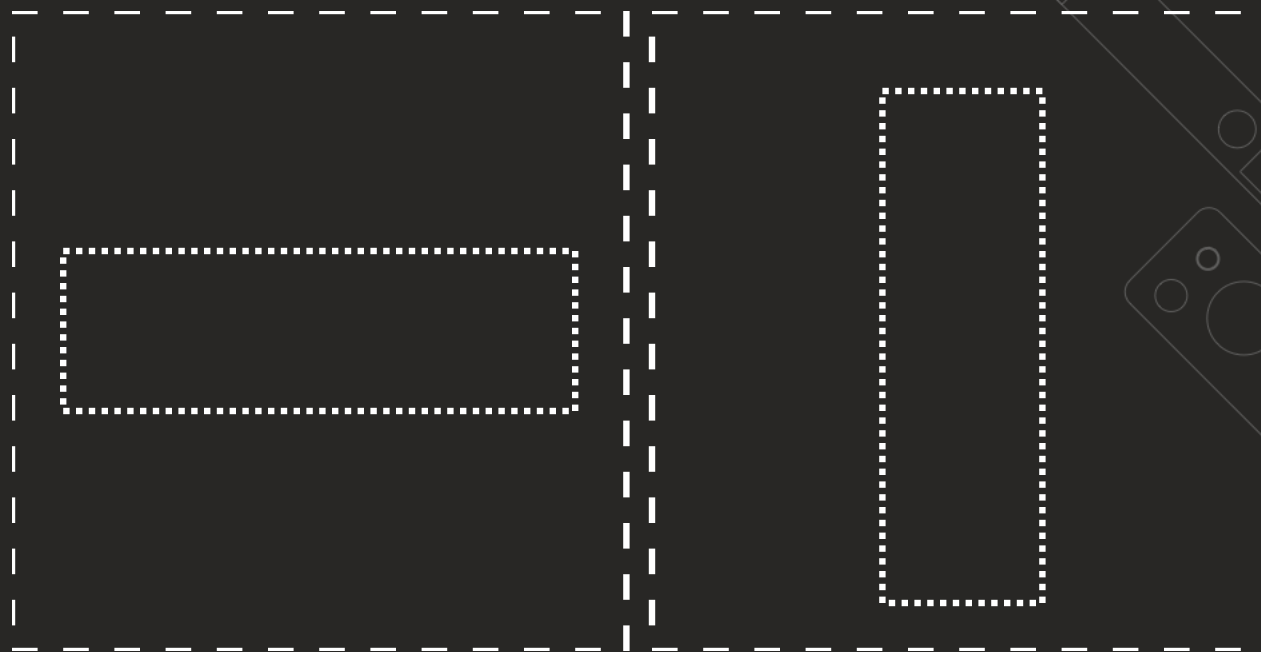
Focus traverser

- Default JUCE keyboard focus traverser goes top-to-bottom, *then* left to right
- You may run into issues depending on how you draw your component within its bounds.

Figure 1: Right-hand component goes first



Figure 2: Left-hand component goes first



Focus order solutions

- Manually set explicit focus order
- Custom `juce::KeyboardFocusTraverser`
 - Possibly overkill for static layouts

or

- Semi-automatically set explicit focus order

Basic fix

```
SomeContainerComponent() {  
    someButton.setExplicitFocusOrder (1); // Starts from 1  
    somePanSlider.setExplicitFocusOrder (2);  
    someLevelSlider.setExplicitFocusOrder (3);  
}
```

A neat little focus order utility

```
SomeContainerComponent() {  
    ExplicitFocusOrder::startingWith (someButton)  
        .then (somePanSlider)  
        .then (someLevelSlider);  
}
```

Behind the scenes

```
class ExplicitFocusOrder {
public:
    static ExplicitFocusOrder startingWith (juce::Component & component) {
        return ExplicitFocusOrder {}.then (component);
    }

    ExplicitFocusOrder & then (juce::Component & component) {
        component.setExplicitFocusOrder (m_index);
        index += 1;
        return *this;
    }
private:
    int index {1};
};
```

Issues

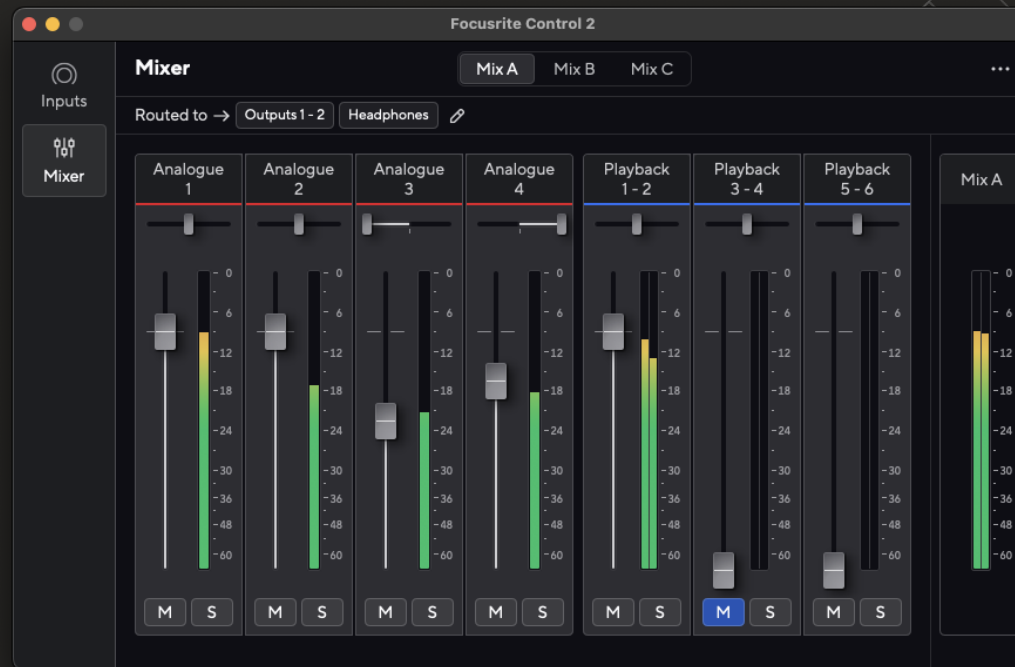
- ✓ Can't interact with the keyboard alone - fixed!
- ✓ Titles missing - fixed!
- ✓ Unclear grouping - fixed!
- ✓ Not clear that buttons can be toggled - fixed!
- ✓ (Windows) Expected the space bar would trigger buttons - fixed!
- ✓ Sliders not receiving keyboard focus - fixed!
- ✓ Strange focus order - fixed!

We did it! 🏁

Further reading & thoughts

- Ed Davies - *Making JUCE Accessible*
youtu.be/BqrEv4ApH3U
- Web views are gaining popularity - *perhaps* consider it
 - Cross-platform accessibility naturally better, due to way content is declaratively "marked up"
 - Tricky decision!
- Sudara's Melatonin Inspector JUCE module
github.com/sudara/melatonin_inspector

Quick tour of Focusrite Control 2



More info about our software stack(s)

Joe Noël - *Stack Overflow: A Tour of the Software Stack at Focusrite*

youtu.be/Xe2r9_CqqkY

More accessibility...

Cain McCormack - *Launchpad iOS: An Accessibility Journey*
youtu.be/ShJun3X1BE4

Thanks for listening 🙌

✉ harry.morley@focusrite.com

🏛 [hmorley](#) @ JUCE Forum

🏛 ADC 2023 Discord

🕸 harrymorley.net

FOCUSRITE GROUP

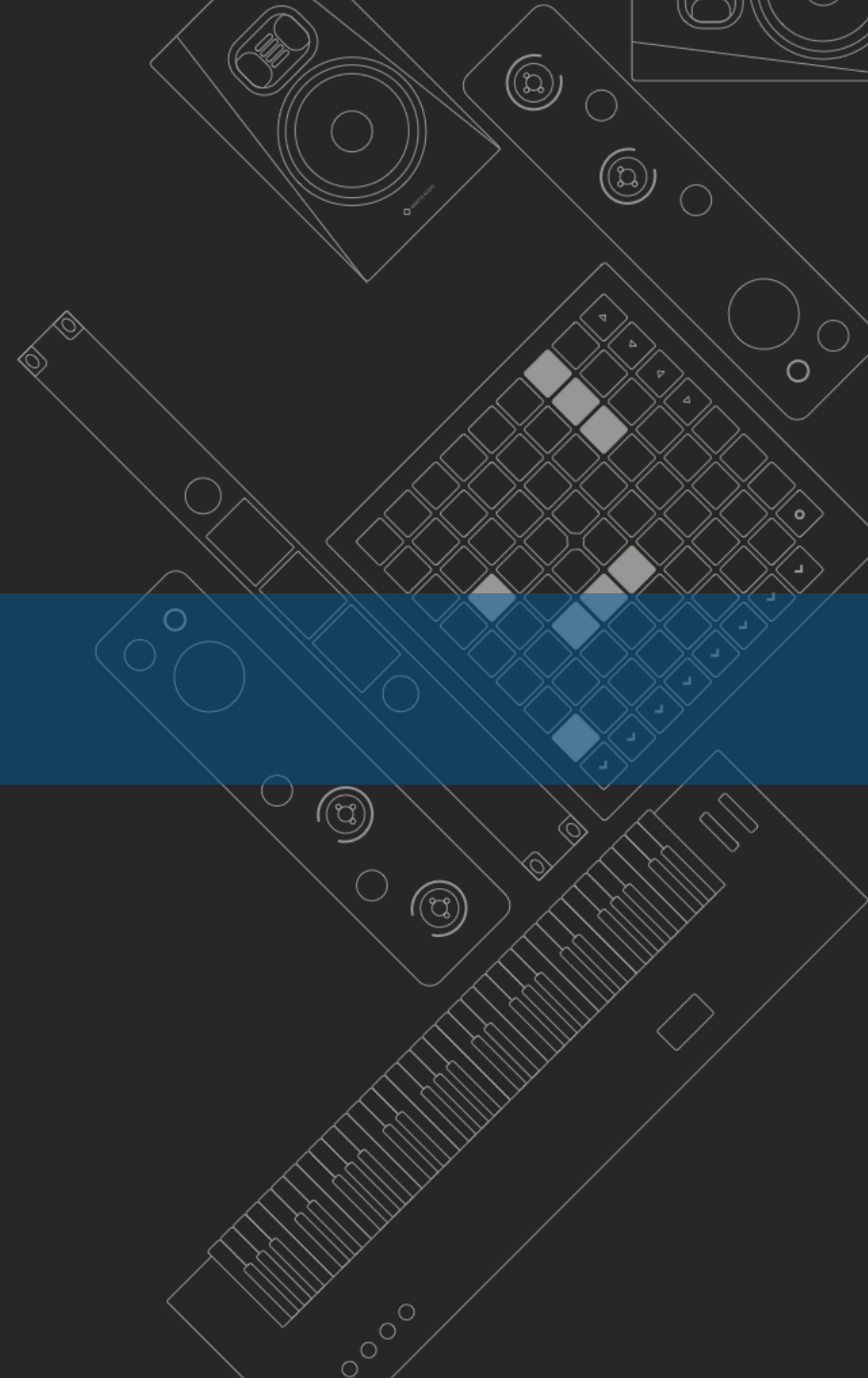
P.S. We're hiring
apply.workable.com/focusrite



P.P.S. Shout-out to James J for the help with the look-and-feel 🙏

And to the rest of the Scarlett team for the support 🙌

Addendum



Accessibility toolkit: basics

- `juce::Component::setAccessible (...)`
- `juce::Component::setTitle (...)`
- `juce::Component::setFocusContainerType (...)`
- `juce::Component::setDescription (...)`
- `juce::Component::setHelpText (...)`

Accessibility toolkit: advanced

- `juce::Component::setExplicitFocusOrder (...)`
- `juce::KeyboardFocusTraverser`
- `juce::Component::createAccessibilityHandler ()`