TELUS Component Library

Button

Multi-Platform Component | Figma UI KIT

Click me

```
<Button onPress={() => alert('Hello world!')}>Click me</Button>
```

Introduction

Buttons allow users to perform an action.

They have multiple styles for various needs, and are ideal for calling attention to where a user needs to do something in order to move forward in a flow.

Follow the appropriate instructions to add this component in to your app.

Guidance

Provide a function as the `onPress` prop to perform an action when clicked.

- Use buttons to move through a transaction
- Aim to use only one primary button variant per page
- Keep the text short and precise. It should fit on a single line (our recommendation is 2-24 characters for French and English)
- Make sure the button text describes an action
- Buttons should not be disabled. Use inline (on blur) error messaging, to provide feedback when the form is invalid

Alternatives

Do not use Button for:

- Navigating to a URL which represents a choice in the current action flow: consider ButtonLink
- One navigation action taking the user to a page that is the only logical next step: consider ChevronLink
- Smaller actions that occur within the current page or screen: consider TextButton
- Navigating to a page outside of the user's current action flow: consider Link

Using refs

`Button` component provides ref forwarding, which allows the user to trigger events like focusing, clicking, etc., programmatically:

Click THIS to focus on THAT

THIS

THAT

```
function buttonRefExample() {
  const ref = useRef()
  const handlePress = () => ref.current.focus()

  return (
    <StackView>
      <Typography>Click THIS to focus on THAT</Typography>
      <Button onPress={handlePress}>THIS</Button>
      <Button ref={ref}>THAT</Button>
    </StackView>
  )
}
```

Accessibility

Insert A11yText any time an interactive element's label lacks context that a sighted user would see in the content around it. Many users of assistive technology explore a page by cycling through the interactive elements first.

The new MegaPhone is our loudest phone yet.

Add to cart

```
<StackView space={2}>
  <Typography>The new MegaPhone is our loudest phone yet.
</Typography>
  <Button variant={{ priority: 'high' }} onPress={() => {}}>
    Add to cart{<A11yText text=", MegaPhone" />}
  </Button>
</StackView>
```

In the above example, screenreaders read "Add to cart, MegaPhone" on focusing the button.

Appending hidden text at the end of the element is the best practice for voice to text technology.

Button is by default described to accessibility tools as having the `button` role. Button accepts React Native's accessibility props (web docs, native docs).

Platform considerations

The component is available on both native platforms and web.

Button background animated transitions are not currently supported in iOS or Android apps. Focus states are not well supported on native platforms.

Props

Name	Type	Platform	Default	Description
tokens	tokens	standard		System tokens prop, see tokens for more details
inactive	bool	standard		If true, prevents the button from being pressed, changes the cursor (on web) and

Name	Type	Platform	Default	Description
				accessibility attributes to communicate this to the user, and applies `inactive: true` appearances from the theme
disabled	bool	standard		Alias for `inactive`
children	custom	standard		Button's children must be either: - One or more text strings and / or A11yText components - A render function with contents that responds to current button state by being passed an object of: ``` { hovered<bool>, focused<bool>, pressed<bool>, inactive<bool>, selected<bool>, textStyles<RNStyle> } ```
onPress	func	standard		Function called when the button is pressed. Required unless the button has a href.
iconProps	exact	standard		Optional variant that may be passed down

Name	Type	Platform	Default	Description
				to the link's icon if there is one
iconPosition	'left' | 'right'	standard		When `icon` is provided, use `iconPosition` to place the Icon to the left or right side of the button.
icon	func	standard		A function component for an SVG icon to render inside the link. Inherits size and color from the link and any Typography the link is nested inside.
variant	variant	standard		System variant prop, see variants for more details

Note: this component also handles all React Native accessibility props

Tokens

In exceptional circumstances, the following tokens can be passed to this component to override its default styles. **Do not do this unless absolutely necessary.** Read more about overriding styles.

▶ View Tokens

Variants

Priority

- Priority indicates the importance of the call-to-action
- Limit the amount of high priority buttons on a page so that you proritize meaningful user flows

High Low

```
<div style={{ display: 'flex', gap: '1em' }}>
  <Button onPress={noop} variant={{ priority: 'high' }}>
    High
  </Button>
  {/* default is low */}
  <Button onPress={noop} variant={{ priority: 'low' }}>
    Low
  </Button>
</div>
```

Danger

- Danger indicates a user action that is destructive that will result in loss of information or cause significant monetary charges
- Limit the use of the danger button whenever possible

Danger

```
<Button onPress={noop} variant={{ danger: true }}>
  Danger
</Button>
```

Size

- Available in default or small size
- Size is not responsive, unless otherwise specified in code

Regular

Small

```
<div style={{ display: 'flex', flexDirection: 'column', gap:
'1rem' }}>
  <Button onPress={noop} variant={{ priority: 'high' }}>
    Regular
  </Button>
  <Button onPress={noop} variant={{ size: 'small', priority:
'high' }}>
    Small
  </Button>
</div>
```

Inverse

- Both high and low priority buttons are available in inverse
- Use inverse buttons on darker backgrounds

High Low

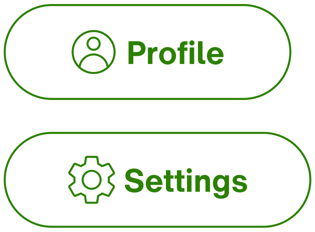
```
<div style={{ display: 'flex', gap: '1em' }}>
  <Button onPress={noop} variant={{ priority: 'high', inverse:
true }}>
    High
  </Button>
  <Button onPress={noop} variant={{ priority: 'low', inverse:
true }}>
    Low
  </Button>
</div>
```

Icons

Icons may be imported and passed to the `icon` prop.

By default, icons are aligned left, ensuring icons align when multiple buttons are stacked vertically.

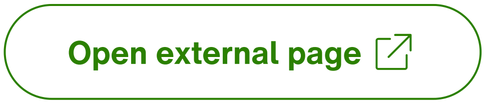
```
import { Profile, Settings } from '@telus-uds/palette-
allium/build/web/icons'
```



```
<StackView space={3}>
  <Button icon={Icons.Profile}>Profile</Button>
  <Button icon={Icons.Settings}>Settings</Button>
</StackView>
```

The icon may be aligned to the right of the text using `iconPosition` prop.

```
import { LinkExternal } from '@telus-uds/palette-
allium/build/web/icons'
```



```
<Button icon={Icons.LinkExternal} iconPosition="right">
  Open external page
</Button>
```

Text

- Use the Text variant to visually deprioritize the Button
- It maintains the same height of Button so that you can put them next to each other without additional effort to vertically align
- Visually appears similar to Link, renders as Button element
- Works with low/high priority and danger variants

High Priority Text Low Priority Text Danger Text

Inverse Text

```
<>
  <div style={{ display: 'flex', gap: '1em' }}>
    <Button onPress={noop} variant={{ text: true, priority:
'high' }}>
      High Priority Text
    </Button>
    <Button onPress={noop} variant={{ text: true, priority:
'low' }}>
      Low Priority Text
    </Button>
    <Button onPress={noop} variant={{ text: true, danger: true
}}>
      Danger Text
    </Button>
  </div>
  <div style={{ backgroundColor: '#000', padding: '10px' }}>
    <Button onPress={noop} variant={{ text: true, inverse: true
}}>
      Inverse Text
    </Button>
  </div>
</>
```

Width

- Available in default, full-width or responsive.
- Default-width - expands based on content.
- Full-width `variant={{ width: 'full'}}` - expands to width of the container.
- Responsive `variant={{ width: 'responsive'}}` - automatically applies full-width when is displayed on the XS viewport.

Regular width

Full width

Responsive width

```
<div style={{ display: 'flex', gap: '1em', flexDirection:
'column' }}>
  <Button>Regular width</Button>
  <Button variant={{ width: 'full' }}>Full width</Button>
  <Button variant={{ width: 'responsive' }}>Responsive
width</Button>
</div>
```

Feedback

- Spotted a problem with this component? Raise an issue on GitHub
- See any existing issues for this component
- Contact the team on slack in #ds-support