

TELUS Component Library

Progress

Multi-Platform Component | [Figma UI KIT](#)



```
<Progress>
  <Progress.Bar percentage={75} accessibilityLabel="Progress
bar" />
</Progress>
```

Introduction

Progress component is a container for displaying one or several

`Progress.Bar` s.

[Follow the appropriate instructions](#) to add this component in to your app.

Accessibility

Both Progress and its child Progress.Bar components accept the standard [React Native accessibility props](#), but these usually don't need to be provided.

The crucial accessibility props are set within Progress.Bar, and defaults are automatically provided as follows:

- `accessibilityRole` (`role`): `progressbar` ,
- `accessibilityValue.min` (`aria-valuemin`): `0` ,
- `accessibilityValue.max` (`aria-valuemax`): `100` ,
- `accessibilityValue.now` (`aria-valuenow`): `percentage` ,
- `accessibilityValue.text` (`aria-valuetext`): `%{percentage}%`

If necessary, these may be overridden by passing these props to Progress.Bar, but for the majority of cases, it automatically generates the expected values for assistive technology to correctly describe the progress bar's current state.

Platform considerations

The component is available on both native platforms and web.

However, keep in mind that appearance may slightly differ among platforms.

Props

Progress

Name	Type	Platform	Default	Description
children*	node	standard		Use the `children` prop to supply one or several `ProgressBar`s.
tokens	tokens	standard		System tokens prop, see tokens for more details
variant	variant	standard		System variant prop, see variants for more details

Progress.Bar

Name	Type	Platform	Default	Description
children	node	standard		Optional children that can be used to customize the progress bar filling.
percentage	number	standard		Percentage of completion.

Name	Type	Platform	Default	Description
tokens	tokens	standard		System tokens prop, see tokens for more details
variant	variant	standard		System variant prop, see variants for more details

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

- Use `{ size: 'mini' }` in the `variant` prop of the Progress component for a narrower progress bar container.
- Use `{ inactive: true }` in the `variant` prop of the Progress.Bar component for a filling corresponding to the disabled state.
- `{ negative: true }` variant of the Progress.Bar component for negative / secondary filling.



```
<Progress variant={{ size: 'mini' }}>
  <Progress.Bar percentage={75} accessibilityLabel="Progress
bar" />
  <Progress.Bar
    percentage={50}
    accessibilityLabel="Negative progress bar"
    variant={{ negative: true }}
  />
```

```
<Progress.Bar  
  percentage={25}  
  accessibilityLabel="Inactive progress bar"  
  variant={{ inactive: true }}  
/>  
</Progress>
```

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](#)