

TELUS Component Library

Carousel

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



Item 1 of 3

```
<Carousel>
  <Carousel.Item>
    
    </Carousel.Item>
  <Carousel.Item>
    
</Carousel.Item>
<Carousel.Item>
  
  </Carousel.Item>
</Carousel>
```

Introduction

A carousel is a component for cycling through slides, designed to take up a designated space to allow the user to interact with the content at their own pace.

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

Guidance

CAUTION

Though the carousel has been built with many accessibility considerations and visual styles tailored to our brand, there are associated risks when deciding to use carousel for your online experiences that include:

- Carousels always include content that is hidden from the customer's view, and it is very likely they will not see that content
 - Carousel items have a low click-through rate
 - Carousels are often skipped entirely by customers
 - Content within carousels are hidden when they are not in view, which may prevent them from being indexed in search results
-
- Use a carousel when other alternatives have been exhausted
 - Users often overlook and scroll past carousel
 - Users often miss content after the first slide
 - Key points or messages can be missed or misinterpreted if they span multiple slides
 - Use a carousel for the following considerations

- Location is a priority and there is limited space
- Provide supplemental content without disrupting the main objective
- Display content in order of priority and relevance
- Must not display content integral to the main objective
- Avoid showing more than 3 panels
- Avoid showing more than 4 slides of full content
- Avoid automatically switching between slides without a user interaction

Usage

- `Carousel` is a top-level export from `@telus-uds/components-base` which is used to render a Carousel
- Immediately within `Carousel`, individual slides are wrapped in `Carousel.Item` for the top-level `Carousel` to know how to identify an individual slide
- You can use any UDS component or other platform-specific component, (based on the platform you're rendering on) to achieve any desired layout
- By default, Carousel takes all the `width` available to it and the `height` is determined based on the content in the slide with more content
- You may want to wrap `Carousel` in other layout components like `Box`, `FlexGrid` etc, to achieve a responsive layout of your need (as shown in the above example)

Mixed content

Carousel may contain mixed content compositions beyond just images, and adapts to the width of the content's overall container.

To avoid repetition, it is often a good idea to set up your content compositions in a subcomponent, to ensure consistency between slides.

This example creates a subcomponent `ContentSlide` and composes slides with an image, a title, descriptive text and a button.



Beat the heat with the coolest deals.

Explore our deals on Unlimited data plans, the latest smartphones, accessories and a whole lot more.

Sign Up



Offer 1 of 3

```
function() {  
  // A content slide to generalize the repeatable slide layout  
  const ContentSlide = ({  
    imageSource,  
    imageAlt,  
    imageSize,
```

```

    boxProps = { horizontal: { xs: 1, md: 5, lg: 8 }},
    headingLabel,
    description,
    slideNumber,
    buttonLabel
  }) => {
    const { activeIndex } = useCarousel()
    const resolvedImageSize = useResponsiveProp(imageSize)
    return (
      <StackView tokens={{ alignItems: 'center' }} space={2}>
        <img src={imageSource} alt={imageAlt}/>
        <Box {...boxProps}>
          <StackView tokens={{ alignItems: 'center' }} space=
{3}>
            <Typography block variant={{ size: 'h1' }}>
              {headingLabel}
            </Typography>
            <Typography>{description}</Typography>
            <Button
              onPress={console.log}
              variant={{ priority: 'high' }}
              tokens={{ alignSelf: 'center' }}
              focusable={activeIndex + 1 === slideNumber}
            >
              {buttonLabel}
            </Button>
          </StackView>
        </Box>
      </StackView>
    )
  }
  return (
    <Carousel refocus itemLabel="offer"
accessibilityLabel="Summer offers">
      <Carousel.Item>
        <ContentSlide
          imageSource="/universal-design-
system/img/lovebirds.jpeg"
          imageAlt="Two colourful love birds sitting together
and connecting."
          imageSize={{ xs: 280, md: 360, lg: 480 }}
          headingLabel="Beat the heat with the coolest deals."
          description="Explore our deals on Unlimited data
plans, the latest smartphones, accessories and a whole lot
more."

```

```

        slideNumber={1}
        buttonLabel="Sign Up"
      />
    </Carousel.Item>
    <Carousel.Item>
      <FlexGrid>
        <FlexGrid.Row>
          <FlexGrid.Col xs={12} sm={6} flex>
            <ContentSlide
              imageUrl="/universal-design-
system/img/guinea-pig.png"
              imageAlt="A guinea pig cooling itself down with
a gentle breeze."
              boxProps={{ left: { xs: 1, md: 3, lg: 5 }}}
              headingLabel="Save $10 monthly"
              description="Example content only."
              slideNumber={2}
              buttonLabel="Sign Up"
            />
          </FlexGrid.Col>
          <FlexGrid.Col xs={12} sm={6} flex>
            <ContentSlide
              imageUrl="/universal-design-
system/img/guinea-pig.png"
              imageAlt="A guinea pig cooling itself down with
a gentle breeze."
              boxProps={{ right: { xs: 1, md: 3, lg: 5 }}}
              headingLabel="Save $10 monthly"
              description="Example content only."
              slideNumber={2}
              buttonLabel="Sign Up"
            />
          </FlexGrid.Col>
        </FlexGrid.Row>
      </FlexGrid>
    </Carousel.Item>
    <Carousel.Item>
      <ContentSlide
        imageUrl="/universal-design-system/img/pigs-allium-
lo.png"
        imageAlt="Some playful pigs with an allium flower."
        imageSize={{ xs: 280, md: 360, lg: 480 }}
        headingLabel="Beat the heat with the coolest deals."
        description="Explore our deals on Unlimited data
plans, the latest smartphones, accessories and a whole lot

```

```
more."
      slideNumber={3}
      buttonLabel="Sign Up"
    />
  </Carousel.Item>
</Carousel>
)
}
```

useCarousel custom hook

```
import { useCarousel } from '@telus-uds/components-base'
const SomeComponentWithinCarouselItem = () => {
  const { activeIndex, totalItems, width, goTo } = useCarousel()
  return <Text>Hi!</Text>
}
```

You can use `useCarousel` to hook into internal state of the Carousel component like:

- `activeIndex`: Index of the current slide
- `totalItems`: Total number of items/slides passed to the Carousel
- `width`: Width of the individual carousel slide
- `goTo`: A function to go to a particular slide by passing the index of that slide, e.g: `goTo(0)` where `0` is the index of the first slide

Tabs navigation

If the `tabs` prop is passed an array of objects with `title` strings and optional additional props, and if the theme considers the viewport to be wide enough, a display of tabs is shown giving a title for each slide.

This is advised for rich-content presentations. Having the title of each item always visible this way increases the chances of a user noticing a relevant item, and helps the user understand the set of items. However, users on narrower screens won't benefit from this.

Get Internet, Optik TV+, SmartHome Security - starting at \$105/month.

Whole Home Bundle helps keep your family connected, entertained, and secure.



Shop bundles



Home bundle for
\$105/month

\$840 off TV & internet
bundles

Save big with the
TELUS Family Discount

```
<Carousel
  itemLabel="offer"
  title="summer offers"
  tabs={ [
    { title: 'Home bundle for $105/month' },
    {
      title: '$840 off TV & internet bundles',
      onPress: () => console.log('Press on second tab'),
      onFocus: () => console.log('Focus in on second tab'),
      onBlur: () => console.log('Focus out on second tab')
    },
    { title: 'Save big with the TELUS Family Discount' }
  ]
}
```



```

    ]]
  >
  <Carousel.Item>
    <Box space={3} bottom={9}>
      <FlexGrid>
        <FlexGrid.Row>
          <FlexGrid.Col flex xs={12} lg={6}>
            <StackView space={3}>
              <Typography variant={{ size: 'h2' }}>
                Get Internet, Optik TV+, SmartHome Security –
starting at $105/month.
              </Typography>
              <Typography>
                Whole Home Bundle helps keep your family
connected, entertained, and secure.
              </Typography>
              <Button variant={{ priority: 'high' }}>Shop
bundles</Button>
            </StackView>
          </FlexGrid.Col>
          <FlexGrid.Col flex xs={12} lg={6}>
            
          </FlexGrid.Col>
        </FlexGrid.Row>
      </FlexGrid>
    </Box>
  </Carousel.Item>
  <Carousel.Item>
    <Box vertical={6}>
      <FlexGrid>
        <FlexGrid.Row>
          <FlexGrid.Col flex xs={12} lg={6}>
            
          </FlexGrid.Col>
          <FlexGrid.Col flex xs={12} lg={6}>
            <StackView space={3}>
              <Typography variant={{ size: 'h2' }}>
                TV and Internet bundles – for up to $840 less!
              </Typography>
              <Typography>

```

```

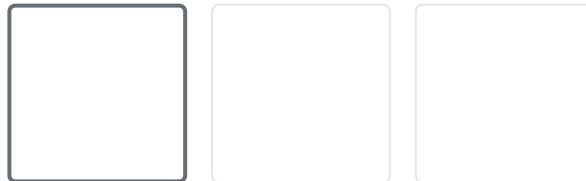
        Get the all the quality content you need for a
        fraction of the price.
      </Typography>
      <Button variant={{ priority: 'high' }}>Shop
bundles</Button>
    </StackView>
  </FlexGrid.Col>
</FlexGrid.Row>
</FlexGrid>
</Box>
</Carousel.Item>
<Carousel.Item>
  <StackView space={3} tokens={{ alignItems: 'center' }}>
    
    <Typography variant={{ size: 'h2' }} tokens={{ alignSelf:
'center' }}>
      Are you eligible for family-sized savings?
    </Typography>
    <Typography>You could save BIG with the TELUS Family
Discount.</Typography>
    <Box>
      <ChevronLink>Find out now</ChevronLink>
    </Box>
  </StackView>
</Carousel.Item>
</Carousel>

```

Thumbnail navigation

You can add thumbnail navigation to the `Carousel` component by passing an array of thumbnail properties via the `thumbnails` prop. Each element of such an array may contain the following properties:

- `src`: image source for the thumbnail, required; note that the thumbnail will be rendered as is (but scaled as per the theme), so please make sure that the images are cropped and optimized accordingly
- `alt`: alternative text to be used in case the image cannot be rendered or on hover
- `accessibilityLabel`: text to be used as an a11y label.



```
<Carousel
  thumbnails=[
    {
      src:
        'https://images.ctfassets.net/3cqlnin176yn/E5Ht0TWwg7IcAqPNB0egr
        /9970531c07fd149a09726c975a0e34db/guinea-pig.png',
      accessibilityLabel: 'A guinea pig'
    },
    {
      src:
        'https://images.ctfassets.net/3cqlnin176yn/E5Ht0TWwg7IcAqPNB0egr
        /9970531c07fd149a09726c975a0e34db/guinea-pig.png?',
      accessibilityLabel: 'Another guinea pig'
    },
    {
```

```

      src:
        'https://images.ctfassets.net/3cqlnin176yn/E5Ht0TWwg7IcAqPNB0egr
        /9970531c07fd149a09726c975a0e34db/guinea-pig.png??',
        accessibilityLabel: 'Yet another guinea pig'
      }
    ]}
  >
  <Carousel.Item>
    
  </Carousel.Item>
  <Carousel.Item>
    
  </Carousel.Item>
  <Carousel.Item>
    
  </Carousel.Item>
</Carousel>

```

Advance usage - combining the box component and Carousel

You can wrap the Carousel within the Box component and set a background variant for the Box and using the `useCarousel` hooks it's possible to control the background color of the box based on the current shown slide, when using tabs you can use `inverse:true` property to make the tabs looks lighter. For the Carousel Step tracker on mobile you need to pass the `stepTrackerVariant` prop and set `inverse` to `true` below there is a complete example.

Lorem ipsum dolor sit amet

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus vitae turpis sit amet nulla faucibus tincidunt in vitae leo. Integer luctus arcu eget rutrum, non aliquam risus tincidunt. Pellentesque suscipit eros ac sem imperdiet, nec consequat eros convallis. Etiam quis pretium magna, quis interdum enim. Ut sed dui tristique, gravida orci ut, aliquam diam. Integer aliquam fermentum ante, a placerat odio ullamcorper eu. Nulla vel velit eu velit pellentesque facilisis.

A label for the button

Feature Primary Feature Secondary Feature Neutral Feature Brand Light Lightest Dark Darkest

```
function() {

  const [currentBackground, setCurrentBackground] =
    React.useState('purpleGradient')
  const [isStepTrackerInverse, setIsStepTrackerInverse] =
    React.useState(false)

  const item = {
    headingLabel: 'Lorem ipsum dolor sit amet',
    description:
      'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus vitae turpis sit amet nulla faucibus tincidunt in vitae leo. Integer luctus arcu eget rutrum, non aliquam risus tincidunt. Pellentesque suscipit eros ac sem imperdiet, nec consequat eros convallis. Etiam quis pretium magna, quis interdum enim. Ut sed dui tristique, gravida orci ut, aliquam diam. Integer aliquam fermentum ante, a placerat odio ullamcorper eu. Nulla vel velit eu velit pellentesque facilisis.',
    buttonLabel: 'A label for the button',
    setCurrentBackground,
    setIsStepTrackerInverse
  }

  const InverseTemplate = ({
    headingLabel,
```

```

description,
buttonLabel,
inverse = true,
index,
setCurrentBackground,
bg,
setIsStepTrackerInverse
}) => {
  const { activeIndex } = useCarousel()

  useEffect(() => {
    if (setCurrentBackground && activeIndex === index && bg) {
      setCurrentBackground(bg)
      setIsStepTrackerInverse(inverse)
    }
  }, [setCurrentBackground, activeIndex, bg, index,
setIsStepTrackerInverse, inverse])

  return (
    <Box space={5}>
      <Box vertical={2} bottom={5}>
        <Typography block variant={{ size: 'h1', inverse }}>
          {headingLabel}
        </Typography>
      </Box>
      <Box bottom={5}>
        <Typography variant={{ inverse }}>{description}
      </Typography>
      </Box>
      <Box bottom={5}>
        <Button onPress={() => {}} variant={{ priority: 'high'
}} tokens={{ alignSelf: 'center' }}>
          {buttonLabel}
        </Button>
      </Box>
    </Box>
  )
}

return (
  <Box space={5} variant={{ background: currentBackground }}>
    <Carousel
      previousNextNavigationPosition='inside'
      previousNextIconSize='default'
      tabs={[

```

```

    { title: 'Feature Primary', inverse: true },
    { title: 'Feature Secondary', inverse: true },
    { title: 'Feature Neutral' },
    { title: 'Feature Brand', inverse: true },
    { title: 'Light' },
    { title: 'Lightest' },
    { title: 'Dark', inverse: true },
    { title: 'Darkest', inverse: true }
  ]
  stepTrackerVariant={{ inverse: isStepTrackerInverse }}
>
  {[
    { ...item, bg: 'featurePrimary' },
    { ...item, bg: 'featureSecondary' },
    { ...item, bg: 'featureNeutral', inverse: false },
    { ...item, bg: 'featureBrand' },
    { ...item, bg: 'light', inverse: false },
    { ...item, bg: 'lightest', inverse: false },
    { ...item, bg: 'dark' },
    { ...item, bg: 'darkest' }
  ].map((itemProps, index) => (
    <Carousel.Item key={` ${item.bg}-carousel`} >
      <InverseTemplate {...itemProps} index={index} />
    </Carousel.Item>
  )))
</Carousel>
</Box>
)
}

```

Accessibility

When implementing the carousel, take care to ensure the component is accessible.

- Must supply accessible content for each slide
- Images must have appropriate “alt” text in English and French
- Must be able to navigate and read content using assistive technologies

Accessibility features



DANGER

Carousel is currently in Beta and many of these are work-in-progress.

To attempt to mitigate the inherent difficulties of using a Carousel with assistive technology, the UDS carousel has the following features:

- (TODO) Renders content as a semantic unordered list
- (TODO) Renders a single slide at a time to the DOM
 - (TO IMPROVE) Mitigates overwhelming a user of assistive technologies by not announcing all content at once
 - (TODO) Allows user to navigate to the next slide within the slider, or skip and navigate away from the slider
- (TODO) Announces each slide to assistive technologies using the "aria-live" attribute on a visually hidden `<div>` element. Example: `<div aria-live="polite">{`Slide number ${currentSlide} of ${total} slides`}</div>`
- (TODO) Each active slide will receive focus so the user could interact with its content
 - (TO IMPROVE) By default, `` elements cannot receive focus
 - (TODO) Set `tabindex` attribute to `-1`; allow the element to receive focus through JavaScript
- (TO IMPROVE) Must have a descriptive text on each navigational element for assistive technologies
 - (TODO) Must allow ability to switch it to French using the required `copy` prop

Platform considerations

The component is available on both native platforms and web

Props

Carousel

Name	Type	Platform	Default
tokens	tokens	standard	
variant	variant	standard	
stepTrackerVariant	objectOf	standard	
children* ::	union	standard	
itemLabel	string	standard	
previousNextNavigationPosition	'inside' 'outside' 'edge'	standard	
previousNextIconSize	'default' 'small'	standard	

Name	Type	Platform	Default
	'large'		
springConfig	object	standard	
thumbnails	arrayOf	standard	
minDistanceForAction	number	standard	
minDistanceToCapture	number	standard	
onIndexChanged	func	standard	

Name	Type	Platform	Default
skipLinkHref	string	standard	
tabs	arrayOf	standard	
refocus	bool	standard	

Name	Type	Platform	Default
panelNavigation	element	standard	
onAnimationStart	func	standard	

Name	Type	Platform	Default
onAnimationEnd	func	standard	
panelNavigationTextDictionary	shape	standard	
tag	'h1' 'h2' 'h3' 'h4' 'h5' 'h6' 'article' 'aside' 'blockquote' 'footer' 'figure' 'form' 'header' 'ul' 'li' 'main'	standard	

Name	Type	Platform	Default
	'nav' 'section' 'label'		

Carousel.Item

Name	Type	Platform	Default	Description
				Set tag container default theme that the carousel will use to render the items
tag	'h1' 'h2' 'h3' 'h4' 'h5' 'h6' 'article' 'aside' 'blockquote' 'footer' 'figure' 'form' 'header' 'ul' 'li' 'main' 'nav' 'section' 'label'	standard	'li'	Define the HTML tag to use for the items in the carousel. The default is 'li'. If the tag is a block-level element, the carousel will wrap the items in a container. If the tag is an inline-level element, the carousel will not wrap the items. The tag must be a valid HTML tag. The tag must not be a form element. The tag must not be a script element. The tag must not be a style element. The tag must not be a meta element. The tag must not be a title element. The tag must not be a base element. The tag must not be a link element. The tag must not be a script element. The tag must not be a style element. The tag must not be a meta element. The tag must not be a title element. The tag must not be a base element. The tag must not be a link element.

Name	Type	Platform	Default	Description
variant	variant	standard		System provided variant model
elementIndex	number	standard		Index of current slide. Does not provide user feedback. Can be used by top content
children*	union	standard		Content slides
setContentBackgroundColor	func	standard		Function to change carousel content background color of slide display

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

Other considerations

- You may want to use the same kind of layout in all your slides to avoid visual and height differences
- `previous` and `next` navigation buttons are automatically removed in `sm` and `xs` viewports, as these smaller viewports offer swipe functionality

Variants

This component does not have any stylistic variants.

Tokens

You can override the following tokens in exceptional circumstances:

- `previousIcon` - Icon of the previous button
- `nextIcon` - Icon of the next button
- `showPreviousNextNavigation` - If you want to show/hide the previous/next navigation
- `showPanelNavigation` - If you want to show/hide the panel navigation
- `spaceBetweenSlideAndPreviousNextNavigation` - Horizontal space between slide and previous/next navigational buttons
- `spaceBetweenSlideAndPanelNavigation` - Vertical space between slide area and panel navigation area

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](#)