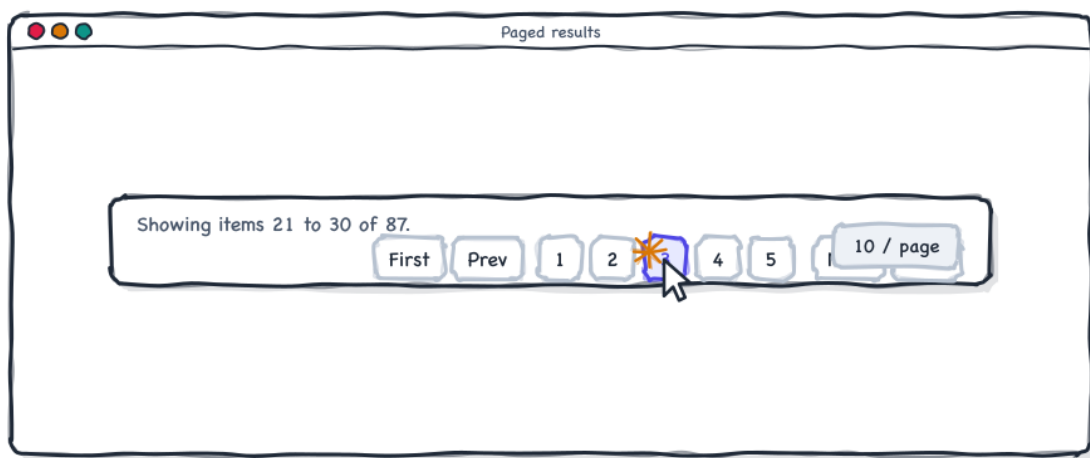


PagingControls

Developer Manual

GemsFX • com.dlsc.gemsfx.paging • JavaFX Custom Controls



The control summarizes the current page and lets the user move through a larger result set.

PagingControls is the standalone pagination control used by the paged list and grid controls. It renders range messages, numeric page indicators, first/previous/next/last navigation and an optional page-size selector. This manual documents its page model, keyboard behaviour, styling and localization hooks.

Table of Contents

1. Introduction	2
1.1 Key features	2
1.2 Maven dependency	2
2. Getting started	2
3. Anatomy	3
4. Control API	4
4.1 Paging model	4
4.2 Button and selector options	4
4.3 Layout and visibility	5
5. Page math and clamping	5
6. Navigation behaviour	5
7. Styling	6
8. Localization	7
9. Accessibility	7
10. Recipes	7
10.1 Hide the page-size selector	7
10.2 Use a wider indicator window	7
10.3 Center the bar	7
10.4 Drive a table	7
10.5 Checklist	8
11. See also	8

1. Introduction

PagingControls is a JavaFX Control for navigating a zero-based paged data set. It does not load data itself; applications or higher-level controls update `totalItemCount`, `pageSize` and `page`.

1.1 Key features

- Zero-based page model with derived `pageCount`.
- Numeric page buttons with configurable maximum count.
- First, previous, next and last navigation.
- Optional page-size selector with configurable choices.
- Range message label and localized button text.
- Keyboard navigation using arrow, page, home and end keys.
- Read-only needed property used by skins to hide unneeded controls.

1.2 Maven dependency

```
<dependency>
  <groupId>com.dlsc.gemsfx</groupId>
  <artifactId>gemsfx</artifactId>
  <version>4.4.1</version>
</dependency>
```

The class is in `com.dlsc.gemsfx.paging`.

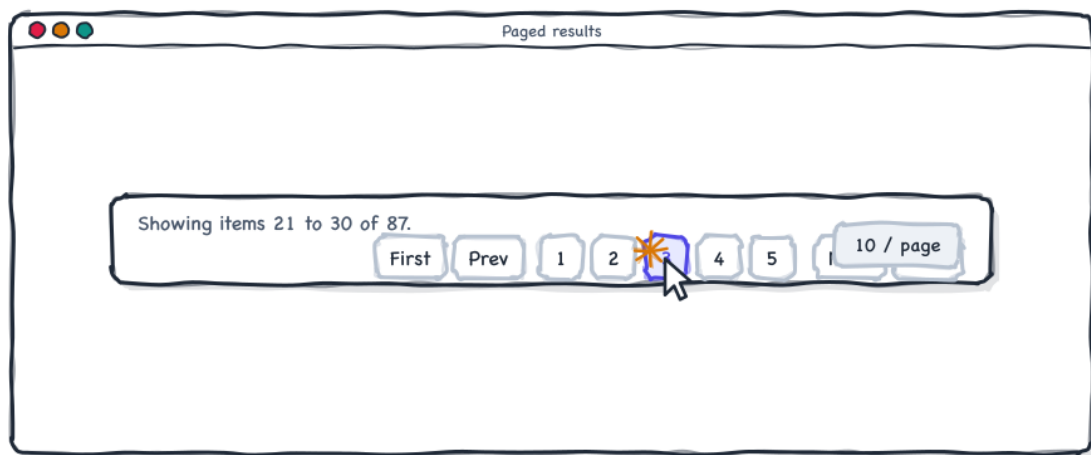
2. Getting started

Use **PagingControls** when the data view is built separately from the navigation bar.

```
PagingControls controls = new PagingControls();
controls.setTotalItemCount(87);
controls.setPageSize(10);

controls.pageProperty().addListener((obs, oldPage, page) -> {
    int offset = page.intValue() * controls.getPageSize();
    loadRows(offset, controls.getPageSize());
});
```

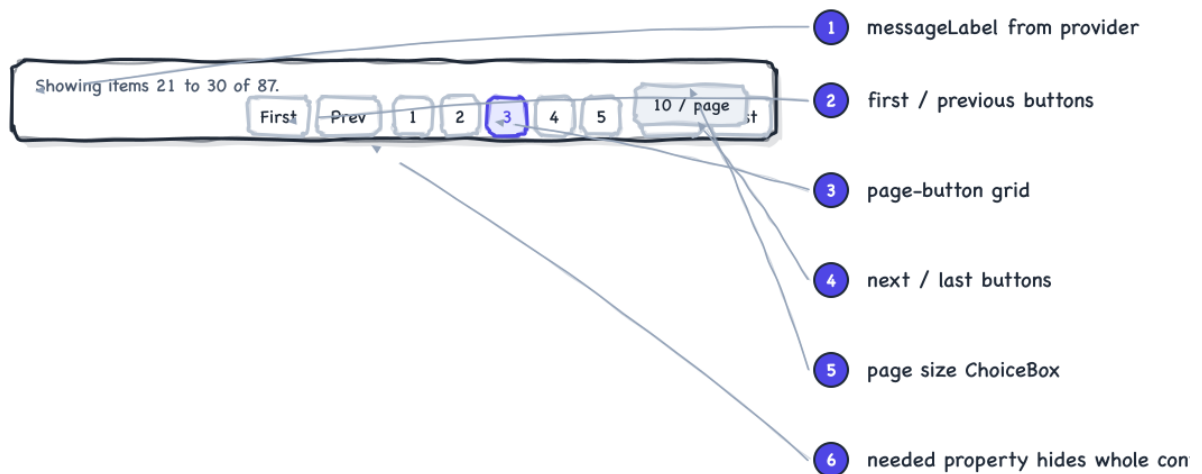
A minimal standalone pager.



PagingControls with a message, navigation buttons and a page-size selector.

3. Anatomy

The skin arranges message, buttons and page-size controls in one pane. LEFT and RIGHT alignment use a horizontal layout; CENTER uses a vertical layout.



The parts of the PagingControls skin.

Part	Style class	Description
.paging-controls	Root style class on PagingControls.	
.pane	Container used by the skin. It also receives horizontal or vertical.	
.message-label	Message label generated from the current range and total count.	

Part	Style class	Description
.buttons-container / .grid-pane	Container for navigation and page buttons.	
.page-button, .current	Numeric page buttons; current marks the selected page.	
.navigation-button	Base class for first, previous, next and last buttons.	
.page-size-container	Container for the page-size label and choice box.	

4. Control API

4.1 Paging model

Property	Type	Default	Description
page	IntegerProperty	0	Zero-based current page. The base class clamps the value when the page count shrinks.
pageSize	IntegerProperty	10	Number of items represented by one page.
totalItemCount	IntegerProperty	0	Total number of items in the backing data set. It drives the derived page count.
pageCount	ReadOnlyIntegerProperty	0	Computed from totalItemCount and pageSize, rounded up.

4.2 Button and selector options

Property	Type	Default	Description
availablePageSizes	ObservableList<Integer>	5, 10, 15, 20, 30, 40, 50	Values shown by the page-size selector.
pageSizeSelectorLabel	StringProperty	Results per page	Label in front of the page-size choice box.
showPageSizeSelector	BooleanProperty	true	Shows the choice box when true. Styleable as -fx-show-page-size-selector.
sameWidthPageButtons	BooleanProperty	false	Gives all numeric page buttons the same width. Styleable as -fx-same-width-page-buttons.
showPreviousNextPageButton	BooleanProperty	true	Shows previous and next navigation buttons. Styleable as -fx-show-previous-next-page-button.
messageLabelStrategy	ObjectProperty<MessageLabelStrategy>	ALWAYS_SHOW	Controls whether the message label is visible. Styleable as -fx-message-label-strategy.
firstLastPageDisplayMode	ObjectProperty<FirstLastPageDisplayMode>	SHOW_PAGE_BUTTONS	Controls first/last page buttons or arrow buttons. Styleable as -fx-first-last-page-display-mode.
maxPageIndicatorsCount	IntegerProperty	5	Maximum number of numeric page indicators. Styleable as -fx-max-page-indicators-count.

4.3 Layout and visibility

Property	Type	Default	Description
alignment	ObjectProperty<HPos>	HPos.RIGHT	Horizontal placement of the controls. Styleable as -fx-page-alignment.
needed	ReadOnlyBooleanProperty	false	True when the skin decides that visible controls are useful for the current page count and options.

5. Page math and clamping

The control computes `pageCount` from `totalItemCount` and `pageSize`. When the total count or page size changes and the current page would be out of range, the current page is clamped.

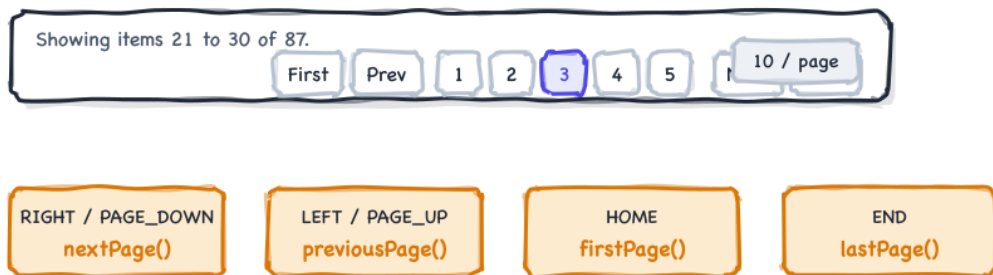


How item count and page size determine the number of pages.

Note. The current page is zero-based. Displayed page buttons are one-based for users.

6. Navigation behaviour

The skin disables navigation buttons that cannot move further. Keyboard handling maps RIGHT and PAGE_DOWN to the next page, LEFT and PAGE_UP to the previous page, HOME to the first page and END to the last page.



Keyboard shortcuts exposed by PagingControls.



First and last pages can be displayed as page buttons, arrow buttons or hidden by display mode.

7. Styling

The user-agent stylesheet is `paging-controls.css`.

CSS property	Type	Default
<code>-fx-show-page-size-selector</code>	boolean	true
<code>-fx-same-width-page-buttons</code>	boolean	false
<code>-fx-show-previous-next-page-button</code>	boolean	true
<code>-fx-message-label-strategy</code>	enum	ALWAYS_SHOW
<code>-fx-first-last-page-display-mode</code>	enum	SHOW_PAGE_BUTTONS
<code>-fx-max-page-indicators-count</code>	integer	5
<code>-fx-page-alignment</code>	HPos	RIGHT

```
.paging-controls {  
    -fx-page-alignment: CENTER;  
    -fx-max-page-indicators-count: 7;  
}  
.paging-controls .page-button.current {  
    -fx-font-weight: bold;  
}
```

Example CSS for a centered pager.

8. Localization

PagingControls resolves its text through ResourceBundleManager.

Key	English default
button.first	First
button.last	Last
button.previous	Previous
button.next	Next
message.items	showing range / total message
label.page-size	Results per page

9. Accessibility

The constructor sets AccessibleRole.PAGINATION. The page buttons are regular JavaFX buttons, so assistive technologies can navigate the same actions as mouse and keyboard users.

10. Recipes

10.1 Hide the page-size selector

```
controls.setShowPageSizeSelector(false);
```

10.2 Use a wider indicator window

```
controls.setMaxPageIndicatorsCount(9);
```

10.3 Center the bar

```
controls.setAlignment(HPos.CENTER);
```

10.4 Drive a table

```
controls.pageProperty().addListener((obs, old, page) -> reloadPage(page.intValue()));
```

10.5 Checklist

- 1 Keep page zero-based in code and one-based in labels.
- 2 Update `totalItemCount` whenever the data source changes.
- 3 Choose `availablePageSizes` that your loader supports.
- 4 Localize button and message text when shipping translated applications.

11. See also

- Demo app: `PagingControlsApp`. Run with `mvn javafx:run -f gemsfx-demo/pom.xml -Dmain.class=com.dlsc.gemsfx.demo.PagingControlsApp`.
- Related controls: `PagingListView` and `PagingGridTableView`.
- API docs: <https://dlsc-software-consulting-gmbh.github.io/GemsFX/api/>