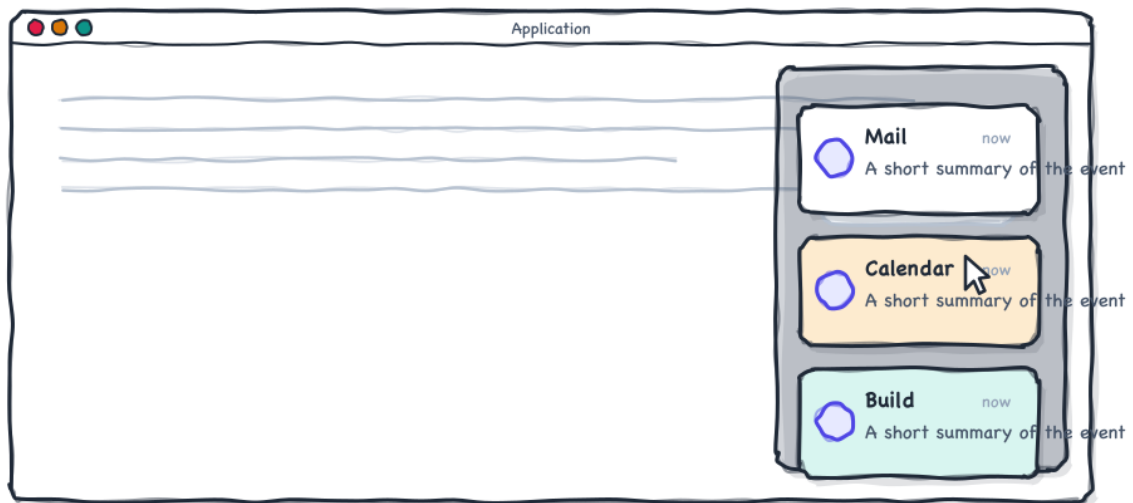


InfoCenterPane

Developer Manual

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



Notifications slide over application content and can be grouped, stacked, pinned and dismissed.

InfoCenterPane is a StackPane that overlays an InfoCenterView on top of application content. The view manages notification groups, stacked notifications, show-all lists, actions, pinning and click behaviours.

Table of Contents

1. Introduction	2
1.1 Key features	2
1.2 Maven dependency	2
2. Getting started	2
3. Anatomy	3
4. Pane API	4
5. InfoCenterView API	4
6. Notification model	5
7. Grouping, stacking and show all	6
8. Pinning and auto-hide	7
9. Actions and click behaviour	7
10. Styling	7
11. Localization	8
12. Accessibility	9
13. Recipes	9
13.1 Pin the center	9
13.2 Change the overlay position	9
13.3 Use a custom view	9
13.4 Clear all notifications	9
13.5 Checklist	9
14. See also	10

1. Introduction

InfoCenterPane extends **StackPane**. It owns one **InfoCenterView** and shows that view as a sliding overlay above the application content.

1.1 Key features

- Notification groups with pinned and unpinned sections.
- Stacked collapsed notifications and expanded group views.
- Show-all mode for groups with many notifications.
- Notification actions and **OnClickBehaviour** results.
- Automatic showing when notifications are added.
- Pinned and auto-hide modes.
- Six overlay positions through **InfoCenterViewPos**.

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.infocenter`.

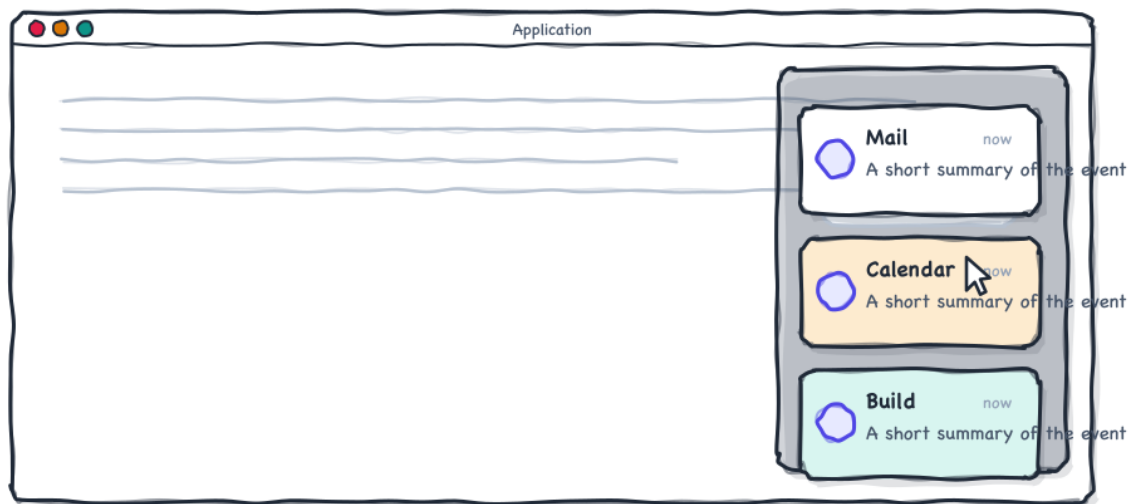
2. Getting started

```
InfoCenterPane pane = new InfoCenterPane();
pane.setContent(applicationRoot);

NotificationGroup<Object, Notification<Object>> group = new NotificationGroup<>("Mail");
pane.getInfoCenterView().getGroups().add(group);

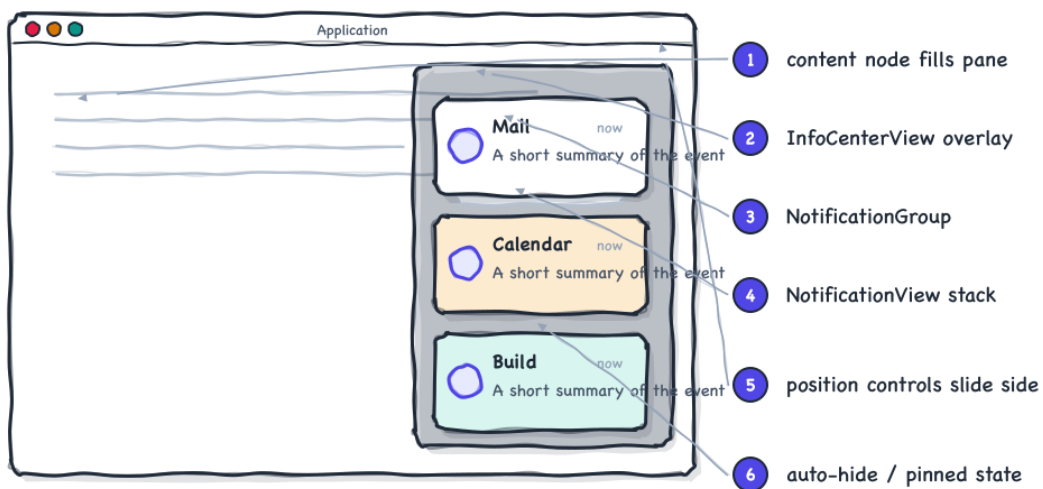
Notification<Object> notification = new Notification<>("New mail", "Quarterly report arrived");
notification.getActions().add(new NotificationAction<>("Open", n -> Notification.OnClickBehaviour.HIDE_AND_REMOVE));
group.getNotifications().add(notification);
```

A pane with one group and one notification.



InfoCenterPane overlays an InfoCenterView on application content.

3. Anatomy



The parts of InfoCenterPane and InfoCenterView.

Part	Description
Content	The application node set with contentProperty.
InfoCenterView	The overlay view owned by the pane.
NotificationGroup	Named list of notifications with sorting, pinning and expansion state.
NotificationView	Visual representation of one notification.
NotificationAction	Action button returning an OnClickBehaviour.

4. Pane API

Property	Type	Default	Description
content	ObjectProperty<Node>	null	Application content filling the InfoCenterPane.
infoCenterView	read-only InfoCenterView	created in constructor	The overlay view containing notification groups.
showInfoCenter	BooleanProperty	false	Shows or hides the overlay.
infoCenterViewPos	ObjectProperty<InfoCenterViewPos>	TOP_RIGHT	Placement: top, center or bottom on the left or right side.
pinned	BooleanProperty	false	Pinned panes remain visible; changing to true shows the pane.
autoHide	BooleanProperty	true	When true, background/content clicks and the timer can hide an unpinned pane.
autoHideDuration	ObjectProperty<Duration>	5 seconds	Time before an unpinned pane hides while the pointer is outside.
slideInDuration	ObjectProperty<Duration>	200 ms	Duration for pane show/hide animation.
insideInfoCenter	ReadOnlyBooleanProperty	false	True while the mouse is inside the InfoCenterView.

Careful. The Java source recommends wrapping InfoCenterPane in another StackPane instead of using it directly as the Scene root on Windows, to avoid occasional platform-specific ghost images.

5. InfoCenterView API

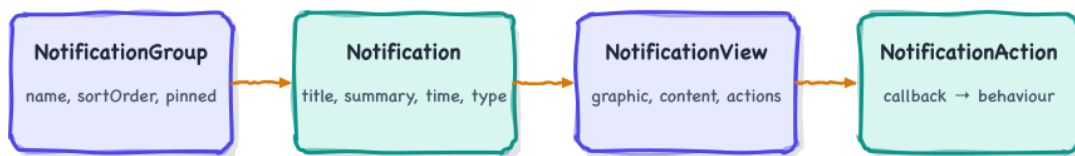
Property	Type	Default	Description
groups	ObservableList<NotificationGroup<?, ?>>	empty list	Groups displayed in the center.
unmodifiableNotifications	ReadOnlyObservableList	flattened list	All notifications across groups.
pinnedGroups / unpinnedGroups	read-only lists	derived	Groups split by pinned state.
autoOpenGroup	BooleanProperty	false	Auto-expands groups. Styleable on InfoCenterView.
showAllGroup	ObjectProperty<NotificationGroup>	null	When set, switches to the single-group list view.
transparent	BooleanProperty	false	Adds style class transparent. Styleable.
placeholder	ObjectProperty<Node>	Label("No notifications")	Displayed when there are no notifications.
notificationSpacing	DoubleProperty	10	Spacing between notification views. Styleable.

Property	Type	Default	Description
showAllFadeDuration / expandDuration / slideInDuration	ObjectProperty<Duration>	300 ms / 300 ms / 500 ms	Animation timings styleable on InfoCenterView.
slideInOrigin	ObjectProperty<Side>	RIGHT	Slide side, bound by InfoCenterPane from its position. Styleable and reflected by :left/:right.

The pane binds the view slide-in origin from the horizontal side of infoCenterViewPos, so left positions slide from the left and right positions slide from the right.

6. Notification model

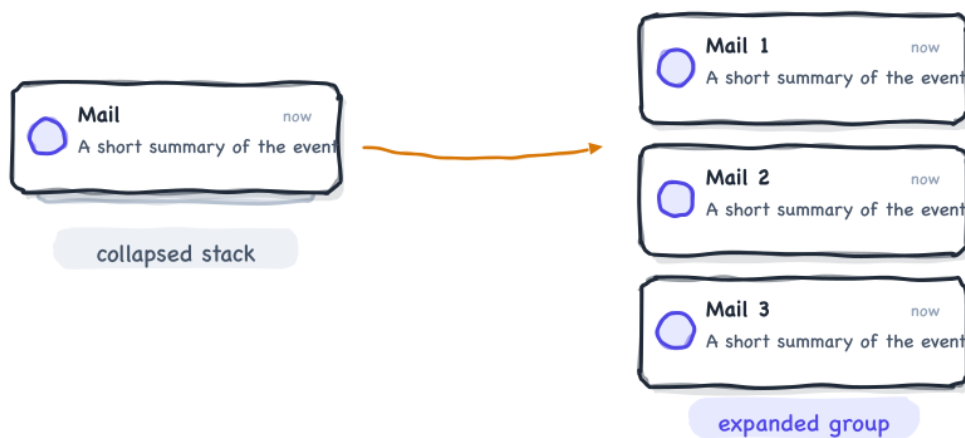
Property	Type	Default	Description
Notification.title	StringProperty	constructor value	Required notification title.
Notification.summary	StringProperty	constructor value	Required summary text.
Notification.dateTime	ObjectProperty<Zoned DateTime>	now or constructor value	Timestamp used for sorting and relative labels.
Notification.type	ObjectProperty<Type>	INFO	INFO, WARNING, ERROR or SUCCESS; affects style classes. ERROR uses danger style.
Notification.onClick	ObjectProperty<Callback>	HIDE_AND_REMOVE	Returns NONE, REMOVE, HIDE or HIDE_AND_REMOVE.
Notification.actions	ObservableList<NotificationAction>	empty list	Buttons shown when a notification is expanded.
NotificationGroup.name	StringProperty	constructor value	Group heading.
NotificationGroup.sortOrder	IntegerProperty	0	Comparable ordering before name fallback.
NotificationGroup.maximumNumberOfNotifications	IntegerProperty	10	Collapsed group limit before show-all is offered.
NotificationGroup.expanded / pinned / pinnable	BooleanProperty	false / false / true	Expansion and pinning state.
NotificationGroup.viewFactory	ObjectProperty<Callback>	new NotificationView(notification)	Factory for custom notification views.



Model objects used by the info center.

7. Grouping, stacking and show all

A collapsed group can render notifications as a stack. Clicking a collapsed group expands it; groups with more items than `maximumNumberOfNotifications` expose a show-all mode backed by a single `ListView`.

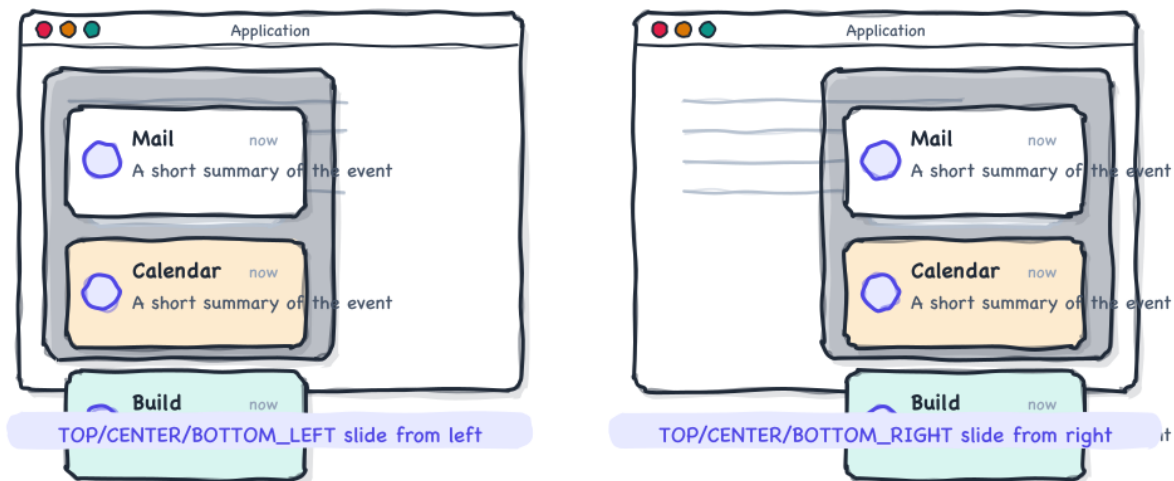


Collapsed stack and expanded group presentation.

- Adding a notification sets its group and binds `notification.expanded` bidirectionally to `group.expanded`.
- `autoCollapse` collapses groups when the notification count drops below two.
- `clearAll()` removes all notifications from all groups.

8. Pinning and auto-hide

The pane shows itself when an `InfoCenterEvent.NOTIFICATION_ADDED` event is fired. It hides on `HIDE` events, background/content clicks or the auto-hide timer when `autoHide` is true and the pane is not pinned.



InfoCenterViewPos controls placement and slide origin.

- `pinned=true` immediately shows the pane.
- When unpinned notifications become empty and the pane is not pinned, it hides.
- Mouse enter and exit maintain `insideInfoCenter` for the auto-hide timer.

9. Actions and click behaviour

Notification clicks and action buttons return an `OnClickBehaviour`. `NONE` leaves everything unchanged, `REMOVE` removes the notification, `HIDE` hides the pane, and `HIDE_AND_REMOVE` does both.

```
notification.setOnClick(n -> Notification.OnClickBehaviour.REMOVE);
notification.getActions().add(new NotificationAction<>("Archive", n -> {
    archive(n.getUserObject());
    return Notification.OnClickBehaviour.HIDE_AND_REMOVE;
}));
```

Custom click and action behaviour.

10. Styling

Style class / pseudo class	Description
<code>.info-center-pane</code>	Root StackPane of the pane.
<code>.info-center-view</code>	Overlay view root.
<code>.main-pane</code> , <code>.top-level-container</code>	Main containers in the view skin.
<code>.groups-container</code> , <code>.group-view</code> , <code>.header</code>	Group containers and headers.
<code>.show-all-button</code> , <code>.show-less-button</code> , <code>.pin-button</code> , <code>.clear-button</code>	Group action buttons.

Style class / pseudo class	Description
.single-group-list-view	ListView used by showAllGroup mode.
.notification-view	Notification view root.
.info, .warning, .danger, .success	Notification type classes.
.stack-notification1, .stack-notification2, .small-stack, .big-stack	Stacking decoration classes.
.transparent	Transparent InfoCenterView variant.
:expanded, :left, :right	Pseudo classes for expanded notification and slide origin.

InfoCenterView CSS property	Default
-fx-auto-open-group	false
-fx-show-all-fade-duration	300ms
-fx-expand-duration	300ms
-fx-slide-in-duration	500ms
-fx-slide-in-origin	RIGHT
-fx-transparent	false
-fx-notification-spacing	10

```
.info-center-view {
    -fx-notification-spacing: 12px;
    -fx-slide-in-origin: RIGHT;
}
.notification-view.warning {
    -fx-font-weight: bold;
}
```

Example CSS for InfoCenterView.

11. Localization

Bundle / key	English default
info-center-view: show-all	Show All
info-center-view: show-less	Show Less
info-center-view: close-group	Close Group
info-center-view: clear-all	Remove all notifications
info-center-view: placeholder	No notifications
info-center-view: accessible.role-description	notifications
notification-view: clear-all	Clear All
notification-view: time.now	now
notification-view: time.hours-ago	h ago

Bundle / key	English default
notification-view: time.minutes-ago	m ago
notification-view: time.yesterday	Yesterday
notification-view: time.days-ago	days ago

12. Accessibility

InfoCenterView sets `AccessibleRole.LIST_VIEW` and uses the localized role description *notifications*. InfoCenterPane itself is a StackPane wrapper around that accessible view.

13. Recipes

13.1 Pin the center

```
infoCenterPane.setPinned(true);
```

13.2 Change the overlay position

```
infoCenterPane.setInfoCenterViewPos(InfoCenterViewPos.BOTTOM_LEFT);
```

13.3 Use a custom view

```
group.setViewFactory(notification -> {  
    NotificationView<Object, MyNotification> view = new NotificationView<>(notification);  
    view.setContent(createDetails(notification));  
    return view;  
});
```

13.4 Clear all notifications

```
infoCenterPane.getInfoCenterView().clearAll();
```

13.5 Checklist

- 1 Create groups first and add them to InfoCenterView.
- 2 Use `sortOrder` for stable group ordering.
- 3 Choose `OnClickBehaviour` deliberately for each notification type.
- 4 Keep pinned groups short by adjusting `maximumNumberOfNotifications`.

14. See also

- Demo app: InfoCenterApp. Run with `mvn javafx:run -f gemsfx-demo/pom.xml -Dmain.class=com.dlsc.gemsfx.demo.InfoCenterApp`.
- Related classes: InfoCenterView, NotificationGroup, Notification, NotificationView and NotificationAction.
- API docs: <https://dlsc-software-consulting-gmbh.github.io/GemsFX/api/>