Adopting New Trackpad Features
Mastering the force
Session 217

Master Raleigh Ledet AppKit Engineer
Force Touch Trackpad
Force Touch Trackpad
Force Touch Trackpad
Force Touch Trackpad
Force Touch Trackpad

Force sensor

Force sensor

Force sensor

Force sensor
Force Touch Trackpad

- Force sensor
- Taptic Engine
- Force sensor
- Force sensor
- Force sensor
- Force sensor
Force Touch Trackpad

Force sensor

Taptic Engine

Force sensor

Force sensor

Force sensor
Force Touch Trackpad

- Force sensor
- Taptic Engine
- Force sensor
- Force sensor
- Force sensor
- Force sensor
Force Touch Trackpad

Taptic Engine

Force sensor

Force sensor
Force Touch Trackpad

Force sensor

Taptic Engine

Force sensor

Force sensor

Force sensor
Force Touch Trackpad
Force Touch Trackpad

Click
Force Touch Trackpad
Force Touch Trackpad

Force Click
Force Touch Trackpad
Force Touch Trackpad
Demo
Force Touch Trackpad
Force Touch Trackpad
Training Schedule

Becoming a master in one day!

Squire

• Accelerator Controls, Table Row Actions

Knight

• Force Event Stream, Spring Loading, Alignment Feedback

Master

• Configuration and Haptics
Squire
Using the built-in tools
Squire Training Schedule

Using the built-in tools

Table Row Actions
Spring Loaded Controls
Accelerator Controls
# Table Row Actions

## Case Study: Mail message list

<table>
<thead>
<tr>
<th>Name</th>
<th>Date</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jeffrey &amp; Diane</td>
<td>5/14/15</td>
<td>Curae duis faucibus accumsan odio curabitur Bill tries to automate the king because the sky is green. Bill gives a sloth to make a pie. Your homie configures my brother for the future of humanity. This cool guy m...</td>
</tr>
<tr>
<td>Roy &amp; Harry</td>
<td>5/14/15</td>
<td>Etiam pretium iaculis justo in hac habitasse platea Bill meets with bill to make a pie.</td>
</tr>
<tr>
<td>Gregory Stone</td>
<td>5/14/15</td>
<td>Metus aenean fermentum donec ut mauris eget massa tempor convallis My favorite spoon explodes a sloth to make a pie. My mate flees from steve to be able to make toast explode. Some guy meets with bill to know more about archeol...</td>
</tr>
</tbody>
</table>
## Table Row Actions

### Case Study: Mail message list

<table>
<thead>
<tr>
<th>Name</th>
<th>Date</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jeffrey &amp; Diane</td>
<td>5/14/15</td>
<td>Curae Duis faucibus accumsan odio curabitur Bill tries to automate the king because the sky is green. Bill gives a sloth to make a pie. Your homie configures my brother for the future of humanity. This cool guy m...</td>
</tr>
<tr>
<td>Roy &amp; Harry</td>
<td>5/14/15</td>
<td>Etiam pretium iaculis justo in hac habitasse platea Bill meets with bill to make a pie.</td>
</tr>
<tr>
<td>Gregory Stone</td>
<td>5/14/15</td>
<td>Metus Aenean fermentum donec ut mauris eget massa tempor convallis My favorite spoon explodes a sloth to make a pie. My mate flees from steve to be able to make toast explode. Some guy meets with bill to know more about archeol...</td>
</tr>
</tbody>
</table>
Table Row Actions

Case Study: Mail message list

NSTableViewDelegate

```swift
optional func tableView(NSTableView, rowActionsForRow: NSInteger, edge: NSTableRowActionEdge) -> [NSTableViewRowAction]
```
Table Row Actions

Case Study: Mail message list

NSTableViewDelegate

optional func tableView(NSTableView,
rowActionsForRow: NSIntegerField, edge: NSTableRowActionEdge)
-> [NSTableViewRowAction]

.Leading
.Trailing
Table Row Actions

Case Study: Mail message list

NSTableViewRowAction
convenience init(style: NSTableViewRowActionStyle, title: String,
handler: ((NSTableViewRowAction, Int) -> Void)
Table Row Actions

Case Study: Mail message list

NSTableViewRowAction

convenience init(style: NSTableViewRowActionStyle, title: String, handler: ((NSTableViewRowAction, Int) -> Void))
**Table Row Actions**

**Case Study: Mail message list**

`NSTableViewRowAction`

convenience `init(style: NSTableViewRowActionStyle, title: String, handler: ((NSTableViewRowAction, Int) -> Void))`

- **Regular**
- **Destructive**
## Case Study: Mail message list

<table>
<thead>
<tr>
<th>Sender</th>
<th>Date</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jeffrey &amp; Diane</td>
<td>5/14/15</td>
<td>Bill tries to automate the king because the sky is green. Bill gives a sloth to make a pie. Your homie configures my brother for the future of humanity. This cool guy m...</td>
</tr>
<tr>
<td>Roy &amp; Harry</td>
<td>5/14/15</td>
<td>Bill meets with Bill to make a pie.</td>
</tr>
<tr>
<td>Gregory Stone</td>
<td>5/14/15</td>
<td>My favorite spoon explodes a sloth to make a pie. My mate flees from steve to be able to make toast explode. Some guy meets with bill to know more about archeol...</td>
</tr>
</tbody>
</table>
## Table Row Actions

### Case Study: Mail message list

<table>
<thead>
<tr>
<th>Name</th>
<th>Date</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jeffrey &amp; Diane</td>
<td>5/14/15</td>
<td>Bill tries to automate the king because the sky is green. Bill gives a sloth to make a pie. Your homie configures my brother for the future of humanity. This cool guy m...</td>
</tr>
<tr>
<td>Roy &amp; Harry</td>
<td>5/14/15</td>
<td>Etiam pretium iaculis justo in hac habitasse platea Bill meets with bill to make a pie.</td>
</tr>
<tr>
<td>Gregory Stone</td>
<td>5/14/15</td>
<td>Metus aenean fermentum donec ut mauris eget massa tempor convallis My favorite spoon explodes a sloth to make a pie. My mate flees from steve to be able to make toast explode. Some guy meets with bill to know more about archeol...</td>
</tr>
</tbody>
</table>
Spring Loaded Controls
Case Study: Finder Drag and Drop Navigation
Spring Loaded Controls
Case Study: Finder Drag and Drop Navigation
Spring Loaded Controls
Case Study: Finder Drag and Drop Navigation

NSButton

var **springLoaded**: Bool
Spring Loaded Controls
Case Study: Finder Drag and Drop Navigation

`var springLoaded: Bool`
Spring Loaded Controls

Case Study: Finder Drag and Drop Navigation

NSSegmentedControl

```swift
var springLoaded: Bool
```
Spring Loaded Controls
Case Study: Finder Drag and Drop Navigation

NSSegmentedControl

var springLoaded: Bool
Accelerator Controls
Case Study: Quicktime Player
Accelerator Controls
Case Study: Quicktime Player
Accelerator Controls
Case Study: Quicktime Player
Accelerator Controls
Case Study: Quicktime Player

NSButton

func setButtonType(NSButtonType)
- AcceleratorButton
- MultiLevelAcceleratorButton
Accelerator Controls

Case Study: Quicktime Player

NSButton

```swift
func setButtonType(NSButtonType)
    .AcceleratorButton
    .MultiLevelAcceleratorButton
```
Accelerator Controls
Case Study: Quicktime Player

NSButton

func setButtonType(NSButtonType)
    .AcceleratorButton
    .MultiLevelAcceleratorButton

var doubleValue: Double
    Range: 0, [1–2]
Accelerator Controls

Case Study: Quicktime Player

NSButton

```swift
func setButtonType(NSButtonType)
    .AcceleratorButton
    .MultiLevelAcceleratorButton
```

```swift
var doubleValue: Double
    Range: [0–5]
```

```swift
var maxAcceleratorLevel: Int
    Range: [1–5]
```
Accelerator Controls

Case Study: Quicktime Player

**NSButton**

```swift
func setButtonType(NSButtonType)
    .AcceleratorButton
    .MultiLevelAcceleratorButton
```

**variables**

```
var doubleValue: Double
    Range: [0–5]

var maxAcceleratorLevel: Int
    Range: [1–5]
```
Accelerator Controls

Case Study: Quicktime Player

NSButton

```swift
func setButtonType(NSButtonType)
    .AcceleratorButton
    .MultiLevelAcceleratorButton

var doubleValue: Double
```

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1X</td>
<td>2X</td>
<td>5X</td>
<td>10X</td>
<td>30X</td>
<td>60X</td>
</tr>
</tbody>
</table>
Accelerator Controls

Case Study: Maps
Accelerator Controls
Case Study: Maps
Accelerator Controls
Case Study: Maps

NSSegmentedControl

var trackingMode: NSSegmentSwitchTracking.MomentaryAccelerator
Accelerator Controls

Case Study: Maps

NSSegmentedControl

var trackingMode: NSSegmentSwitchTracking
    .MomentaryAccelerator
Accelerator Controls

Case Study: Maps

NSSegmentedControl

var trackingMode: NSSegmentSwitchTracking.
MomentaryAccelerator

func doubleValueForSelectedSegment() -> Double
Range: 0, [1–2]
Accelerator Controls

Case Study: Photos
Accelerator Controls
Case Study: Photos
Accelerator Controls

Case Study: Photos

NSControl

var continuous: Bool
Accelerator Controls

Case Study: Photos

NSControl

var continuous: Bool
Accelerator Controls
NSButton, NSSegmentControl

- Accelerated
  - Maps: +/- zoom buttons

- Continuous
  - Photos: previous / next photo button

- Time
Accelerator Controls
NSButton, NSSegmentedControl

Accelerated
Maps: +/- zoom buttons
Continuous
Photos: previous / next photo button

Time
Accelerator Controls

NSButton, NSSegmentedControl

Accelerated

Maps: +/- zoom buttons

Continuous

Photos: previous / next photo button

Time
Accelerator Controls

NSButton, NSSegmentedControl

- Accelerated
  - Maps: +/- zoom buttons

- Continuous
  - Photos: previous / next photo button

Time
Squire Summary

Using the built-in tools

Table Row Actions
Spring Loaded Controls
Accelerator Controls
Knight

Understanding how the force flows
Knight Training Schedule

Understanding how the force flows

Force event stream
Spring loading protocol
Alignment feedback
Event Stream
## Event Stream

<table>
<thead>
<tr>
<th>NSLeftMouseDown</th>
<th>NSEventTypeMagnify</th>
<th>NSRightMouseDown</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSLeftMouseDragged</td>
<td>NSEventTypeRotate</td>
<td>NSRightMouseDragged</td>
</tr>
<tr>
<td>NSLeftMouseUp</td>
<td></td>
<td>NSRightMouseUp</td>
</tr>
<tr>
<td>NSMouseEntered</td>
<td>NSKDown</td>
<td>NSScrollWheel</td>
</tr>
<tr>
<td>NSMouseExited</td>
<td>NSKUp</td>
<td>NSFlagsChanged</td>
</tr>
<tr>
<td>Event Type</td>
<td>Event Type</td>
<td>Event Type</td>
</tr>
<tr>
<td>----------------------------</td>
<td>----------------------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>NSLeftMouseDown</td>
<td>NSEvent_subtype</td>
<td>NSRightMouseDown</td>
</tr>
<tr>
<td>NSLeftMouseDragged</td>
<td>NSEvent_subtype</td>
<td>NSRightMouseDragged</td>
</tr>
<tr>
<td>NSLeftMouseUp</td>
<td>NSEvent_subtype</td>
<td>NSRightMouseUp</td>
</tr>
<tr>
<td>NSMouseEntered</td>
<td>NSKeyDown</td>
<td>NSScrollWheel</td>
</tr>
<tr>
<td>NSMouseExited</td>
<td>NSKeyUp</td>
<td>NSFlagsChanged</td>
</tr>
</tbody>
</table>
## Event Stream

### Pressure Gesture

<table>
<thead>
<tr>
<th>Event Mask</th>
<th>Event Mask</th>
<th>Event Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSLeftMouseDownMask</td>
<td>NSEventMaskMagnify</td>
<td>NSRightMouseDownMask</td>
</tr>
<tr>
<td>NSLeftMouseDraggedMask</td>
<td>NSEventMaskRotate</td>
<td>NSRightMouseDraggedMask</td>
</tr>
<tr>
<td>NSLeftMouseUpMask</td>
<td><strong>NSEventMaskPressure</strong></td>
<td>NSRightMouseUpMask</td>
</tr>
<tr>
<td>NSMouseEnteredMask</td>
<td>NSKeyDownMask</td>
<td>NSScrollWheelMask</td>
</tr>
<tr>
<td>NSMouseExitedMask</td>
<td>NSKeyUpMask</td>
<td>NSFlagsChangedMask</td>
</tr>
</tbody>
</table>
trackEventsMatchingMask(NSEventMask,...)

[NSLeftMouseDraggedMask, NSLeftMouseUpMask, NSEventMaskPressure]
Event Stream
Pressure Gesture

trackEventsMatchingMask(NSEventMask,...)

[.LeftMouseDraggedMask, .LeftMouseUpMask, .EventMaskPressure]
trackEventsMatchingMask(NSEventMask,...)

[.LeftMouseDraggedMask, .LeftMouseUpMask, .EventMaskPressure]

func pressureChangeWithEvent(NSEvent)
Event Stream
Pressure Gesture

NSEventTypePressure
NSEventMaskPressure

var phase: NSEventPhase { get }
Event Stream
Pressure Gesture

NSEventTypePressure
NSEventMaskPressure
var phase: NSEventPhase { get }
var stage: Int { get }
Event Stream

Pressure Gesture

NSEventTypePressure
NSEventMaskPressure
var phase: NSEventPhase { get }
var stage: Int { get }

1 – click
Event Stream
Pressure Gesture

NSEventTypePressure
NSEventMaskPressure
var phase: NSEventPhase { get }
var stage: Int { get }
  1 - click
Event Stream
Pressure Gesture

NSEventTypePressure
NSEventMaskPressure
var phase: NSEventPhase { get }
var stage: Int { get }
1 - click
2 - force click
Event Stream
Pressure Gesture

NSEventTypePressure
NSEventMaskPressure

var phase: NSEventPhase { get }

var stage: Int { get }

1 – click
2 – force click
Event Stream
Pressure Gesture

NSEventTypePressure
NSEventMaskPressure

var phase: NSEventPhase { get }
var stage: Int { get }

1 - click
2 - force click
0 - gesture release
Event Stream
Pressure Gesture

NSEventTypePressure
NSEventMaskPressure

var phase: NSEventPhase { get }
var stage: Int { get }

1 - click
2 - force click
0 - gesture release
Event Stream
Pressure Gesture

NSEventTypePressure

var **pressure**: Float { get }
(of the current stage)
Range: [0–1]
Event Stream
Pressure Gesture

NSEventTypePressure

var `pressure`: Float { get }
(of the current stage)
Range: [0–1]
Event Stream

Pressure Gesture

NSEventTypePressure

var pressure: Float { get }
(of the current stage)
Range: [0–1]
Event Stream
Pressure Gesture

NSEventTypePressure

var pressure: Float { get }
var stageTransition: CGFloat { get }
    Range: [0-1]
Event Stream

Pressure Gesture

NSEventTypePressure

var pressure: Float { get }

var stageTransition: CGFloat { get }

Range: [0–1]
Event Stream
Pressure Gesture

NSEventTypePressure

var pressure: Float { get }
var stageTransition: CGFloat { get }
Range: [0–1]
Event Stream

Parallel Mouse and Pressure event streams
Event Stream

Parallel Mouse and Pressure event streams

User Action

Force on trackpad

Force Click threshold

Click threshold

Events

Mouse:

Stage:

Time

Pressure
Event Stream
Parallel Mouse and Pressure event streams

User Action

Events

Mouse:
Stage:

Pressure

Time

Click threshold

Force Click threshold
Event Stream
Parallel Mouse and Pressure event streams

User Action

Events
Mouse:
Stage:

Force on trackpad

Force Click threshold
Click threshold

Pressure

Time

1
1
0
Event Stream

Parallel Mouse and Pressure event streams

User Action

Force on trackpad

Force Click threshold

Click threshold

Events

Mouse:

Stage:

Pressure

Time
Event Stream

Parallel Mouse and Pressure event streams

User Action

Events

Mouse: Move
Stage: 1
Pressure: 1

Time

Force on trackpad
Force Click threshold
Click threshold
Event Stream
Parallel Mouse and Pressure event streams

User Action

Events

Mouse:  Move
Stage: -

Force on trackpad
Force Click threshold
Click threshold

Pressure

Time
Event Stream

Parallel Mouse and Pressure event streams

User Action

Time

Down

Move

1

Stage:

Events

Mouse:

Force on trackpad

Force Click threshold

Click threshold

Pressure

0

1

Time
Event Stream
Parallel Mouse and Pressure event streams
Event Stream

Parallel Mouse and Pressure event streams

User Action

Events

Mouse: Move Down Dragged
Stage: 1

Time

Force on trackpad

Force Click threshold

Click threshold

Pressure

0 1
Event Stream

Parallel Mouse and Pressure event streams

User Action

Events

Mouse: Move Down Dragged
Stage: 1 2

Force on trackpad

Force Click threshold

Click threshold

Pressure

Time
Event Stream

Parallel Mouse and Pressure event streams

User Action

Events

Mouse: Move Down Dragged
Stage: 1 2

Force on trackpad

Force Click threshold

Click threshold

Pressure

Time
Event Stream
Parallel Mouse and Pressure event streams

User Action

Events
Mouse: Move Down Dragged
Stage: - 1 2

Force on trackpad

Pressure

Time

Force Click threshold
Click threshold
Event Stream

Parallel Mouse and Pressure event streams

User Action

Events

Mouse: Move Down Dragged
Stage: - 1 2

Force on trackpad

Force Click threshold

Click threshold

Pressure

Time
Event Stream

Parallel Mouse and Pressure event streams

User Action

Events

<table>
<thead>
<tr>
<th>Mouse:</th>
<th>Move</th>
<th>Down</th>
<th>Dragged</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stage:</td>
<td>-</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Force on trackpad

- Force Click threshold
- Click threshold

Pressure

Time
Event Stream

Parallel Mouse and Pressure event streams

User Action

<table>
<thead>
<tr>
<th>Events</th>
<th>Mouse:</th>
<th>Move</th>
<th>Down</th>
<th>Dragged</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stage:</td>
<td>-</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Force on trackpad

- Force Click threshold
- Click threshold

Pressure

Time
Event Stream

Parallel Mouse and Pressure event streams

User Action

Events
Mouse: Move Down Dragged
Stage: 1 2 1

Force on trackpad
Force Click threshold
Click threshold

Pressure
Time
Event Stream

Parallel Mouse and Pressure event streams

User Action

Events

Mouse:
Stage:

Move
Down
Dragged

Force on trackpad

Force Click threshold

Click threshold

Pressure

Time
Event Stream
Parallel Mouse and Pressure event streams

User Action

Events
Mouse: Move Down Dragged
Stage: 1 2 1

Force on trackpad

Force Click threshold

Click threshold

Pressure

Time
Event Stream

Parallel Mouse and Pressure event streams

User Action

Events Mouse: Move Down Dragged
Stage: - 1 2 1

Force on trackpad

Force Click threshold

Click threshold

Pressure

Time
Event Stream

Parallel Mouse and Pressure event streams

User Action

<table>
<thead>
<tr>
<th>Events</th>
<th>Mouse:</th>
<th>Move</th>
<th>Down</th>
<th>Dragged</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stage:</td>
<td></td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Force on trackpad

Force Click threshold

Click threshold

Pressure

Time
Event Stream

Parallel Mouse and Pressure event streams

User Action

Force on trackpad

Events

Mouse: Move Down Dragged
Stage: - 1 2 1

Pressure

Time

Force Click threshold
Click threshold
Event Stream

Parallel Mouse and Pressure event streams

User Action

Force on trackpad

Force Click threshold

Click threshold

Events

Mouse: Move  Down  Dragged
Stage:  -  1  2  1

Pressure

0  1  1  0

Time
Event Stream

Parallel Mouse and Pressure event streams

User Action

Events

- Mouse: Move, Down, Dragged
- Stage: 1, 2, 1

Time

Force on trackpad

Pressure

Force Click threshold

Click threshold

0

1

1
Event Stream

Parallel Mouse and Pressure event streams

User Action

Force on trackpad

Force Click threshold

Click threshold

Events

Mouse: Move Down Dragged
Stage: - 1 2

Pressure

Time

User Action

Force on trackpad

Force Click threshold

Click threshold

Events

Mouse: Move
Stage: Move

Pressure

Time
Event Stream
Parallel Mouse and Pressure event streams

User Action

Events

Mouse: Move Down Dragged
Stage: 1 2

Force on trackpad
Force Click threshold
Click threshold

Pressure

Time
Event Stream
Parallel Mouse and Pressure event streams

var associatedEventMask: NSEventMask { get }
var associatedEventMask: NSEventMask { get }

if ((mouseDown.associatedEventMask.contains(.EventMaskPressure)) {
    // Pressure Capable Device!
}

Event Stream

Summary

NSEventTypePressure
NSEventMaskPressure

var phase: NSEventPhase { get }

var pressure: Float { get }

var stage: Int { get }

var stageTransition: CGFloat { get }

var associatedEventMask: NSEventMask { get }

func pressureChangeWithEvent(NSEvent)
Spring Loading

@protocol NSSpringLoadingDestination
Spring Loading

@protocol NSSpringLoadingDestination
required (One of these)
func springLoadingEntered(NSDraggingInfo) -> NSSpringLoadingOptions
- and / or -
func springLoadingUpdated(NSDraggingInfo) -> NSSpringLoadingOptions
@protocol NSSpringLoadingDestination
required (One of these)
func springLoadingEntered(NSDraggingInfo) -> NSSpringLoadingOptions
- and / or -
func springLoadingUpdated(NSDraggingInfo) -> NSSpringLoadingOptions

.Disabled
.Enabled
@protocol NSSpringLoadingDestination
required (One of these)
func springLoadingEntered(NSDraggingInfo) -> NSSpringLoadingOptions
  - and / or -
func springLoadingUpdated(NSDraggingInfo) -> NSSpringLoadingOptions

  .Disabled
  .Enabled
  .ContinuousActivation
@protocol NSSpringLoadingDestination
required (One of these)
func springLoadingEntered(NSDraggingInfo) -> NSSpringLoadingOptions
    - and / or -
func springLoadingUpdated(NSDraggingInfo) -> NSSpringLoadingOptions

. Disabled
. Enabled
. ContinuousActivation
. NoHover
Spring Loading

@protocol NSSpringLoadingDestination
required (One of these)
func springLoadingEntered(NSDraggingInfo) -> NSSpringLoadingOptions
- and / or -
func springLoadingUpdated(NSDraggingInfo) -> NSSpringLoadingOptions

@optional
func springLoadingExited(NSDraggingInfo)
func draggingEnded(NSDraggingInfo)
@protocol NSSpringLoadingDestination
@required
func springLoadingActivated(_ BOOL: Bool,DraggingInfo: NSDraggingInfo)
func springLoadingHighlightChanged(DraggingInfo: NSDraggingInfo)
@protocol NSSpringLoadingDestination
@required
func springLoadingActivated(Bool, draggingInfo: NSDraggingInfo)
func springLoadingHighlightChanged(NSDraggingInfo)

- draggingInfo.springLoadingHighlight
- NSSpringLoadingHighlight
  - None
  - Standard
  - Emphasized
Spring Loading

NSSpringLoadingDestination
Alignment Feedback
Alignment Feedback
Alignment Feedback

Tracking Loop
Alignment Feedback

```
func mouseDown()

nextEvent

move Item

MouseUp?

setNeedsDisplay()
```

Tracking Loop

eventMask
Alignment Feedback

func mouseDown()

nextEvent

move Item

Tracking Loop

eventMask

MouseUp?

NSAlignmentFeedbackFilter

setNeedsDisplay()
class func `inputEventMask()` -> NSEventMask
Alignment Feedback

- **func mouseDown()**
- **nextEvent**
- **move Item**
- **eventMask**

**Tracking Loop**

**MouseUp?**

**NSAlignmentFeedbackFilter**

**setNeedsDisplay()**
func mouseDown()

nextEvent

move Item

NSAlignmentFeedbackFilter

MouseUp?

setNeedsDisplay()

func updateWithEvent(NSEvent)
func mouseDown()

nextEvent

move Item

NSAlignmentFeedbackFilter

MouseUp?

setNeedsDisplay()

func updateWithPanRecognizer(NSPanGestureRecognizer)
func mouseDown()

nextEvent

move Item

Tracking Loop

eventMask

MouseUp?

NSAlignmentFeedbackFilter

setNeedsDisplay()
Alignment Feedback

- func mouseDown()
- prepareAlignment
- move Item
- nextEvent
- Tracking Loop
- MouseUp?
- NSAlignmentFeedbackFilter
- setNeedsDisplay()
Alignment Feedback

func mouseDown()

nextEvent

move Item

prepareAlignment

NSAlignmentFeedbackFilter

MouseUp?

setNeedsDisplay()
Alignment Feedback

Preparing alignment
Alignment Feedback

Preparing alignment
Alignment Feedback
Preparing alignment

default

previous
Alignment Feedback
Preparing alignment

previous

default

previous
Alignment Feedback

Preparing alignment

previous

default

previous
func alignmentFeedbackTokenForHorizontalMovementInView(view: NSView?,
    previousX: CGFloat, alignedX: CGFloat, defaultX: CGFloat) -> NSAditionalFeedbackToken?
func alignmentFeedbackTokenForHorizontalMovementInView(itemView: NSView?,
    previousX: CGFloat, alignedX: CGFloat, defaultX: CGFloat)
    -> NSAlignmentFeedbackToken?

func alignmentFeedbackTokenForVerticalMovementInView(itemView: NSView?,
    previousY: CGFloat, alignedY: CGFloat, defaultY: CGFloat)
    -> NSAlignmentFeedbackToken?
Alignment Feedback
Preparing alignment

```swift
func alignmentFeedbackTokenForHorizontalMovementInView(view: NSView?,
            previousX: CGFloat, alignedX: CGFloat, defaultX: CGFloat)
            -> NSAlignmentFeedbackToken?

func alignmentFeedbackTokenForVerticalMovementInView(view: NSView?,
            previousY: CGFloat, alignedY: CGFloat, defaultY: CGFloat)
            -> NSAlignmentFeedbackToken?
```
func alignmentFeedbackTokenForMovementInView(NSView?,
    previousPoint: NSPoint, alignedPoint: NSPoint, defaultPoint: NSPoint)
-> NSA1ignmentFeedbackToken?
Alignment Feedback

func mouseDown()

eventMask

nextEvent

MouseUp?

move Item

NSAlignmentFeedbackFilter

prepareAlignment

setNeedsDisplay()
Alignment Feedback

func mouseDown()

nextEvent

move Item

prepareAlignment

<Token>

NSAlignmentFeedbackFilter

setNeedsDisplay()

MouseUp?

Tracking Loop

eventMask

nextEvent
Alignment Feedback

func mouseDown()

nextEvent

move Item

prepareAlignment

[<Token>]

NSAlignmentFeedbackFilter

performFeedback

setNeedsDisplay()

MouseUp?

Tracking Loop

eventMask
func performFeedback([<NSAlignmentFeedbackToken>],
performanceTime: NSHapticFeedbackPerformanceTime)
Alignment Feedback

Feeling it

Consistent feel across apps
Easy to integrate into tracking loops
Dragging, resizing, etc…
Knight Summary

Understanding how the force flows

Force event stream
Spring loading protocol
Alignment Feedback
Master
Controlling the force
Master Training Schedule

Controlling the force

Trackpad configuration
Haptics
Trackpad Configuration
More than just Force Click

NSPressureConfiguration
Trackpad Configuration
More than just Force Click

NSPressureConfiguration

init(pressureBehavior: NSPressureBehavior)
Trackpad Configuration
More than just Force Click

NSPressureConfiguration

init(pressureBehavior: NSPressureBehavior)

.PrimaryDefault
.PrimaryClick
.PrimaryGeneric
.PrimaryAccelerator
.PrimaryDeepClick
.PrimaryDeepDrag
Trackpad Configuration

More than just Force Click

NSPressureConfiguration

init(pressureBehavior: NSPressureBehavior)
func set()
Trackpad Configuration
More than just Force Click

NSPressureConfiguration

init(pressureBehavior: NSPressureBehavior)
func set()

Set is not ideal
• Only valid during a drag
• Racing the user
Trackpad Configuration
More than just Force Click

Preferred approach:

```swift
NSView - var pressureConfiguration: NSPressureConfiguration
```

- Configured before mouse down
- Configured when app not responsive
Haptics
The Trackpad Strikes Back
Haptics
The Trackpad Strikes Back

Use sparingly
Subtle interactions
Haptics
The Trackpad Strikes Back

NSHapticFeedbackManager
class func defaultPerformer() -> <NSHapticFeedbackPerformer>

@protocol NSHapticFeedbackPerformer <NSObject>
func performFeedbackPattern(NSHapticFeedbackPattern,
                           performanceTime: NSHapticFeedbackPerformanceTime)
Haptics

The Trackpad Strikes Back

```swift
NSHapticFeedbackManager
class func defaultPerformer() -> <NSHapticFeedbackPerformer>

@protocol NSHapticFeedbackPerformer <NSObject>
func performFeedbackPattern(NSHapticFeedbackPattern,
                           performanceTime: NSHapticFeedbackPerformanceTime)
@end
```
Haptics
The Trackpad Strikes Back

NSHapticFeedbackManager
class func defaultPerformer() -> <NSHapticFeedbackPerformer>

@protocol NSHapticFeedbackPerformer <NSObject>
func performFeedbackPattern(NSHapticFeedbackPattern,
    performanceTime: NSHapticFeedbackPerformanceTime)
@end

NSHapticFeedbackPattern
.Generic
.Alignment
.LevelChange
Haptics
The Trackpad Strikes Back

NSHapticFeedbackManager
class func defaultPerformer() -> <NSHapticFeedbackPerformer>

@protocol NSHapticFeedbackPerformer <NSObject>
func performFeedbackPattern(NSHapticFeedbackPattern,
 performanceTime: NSHapticFeedbackPerformanceTime)
@end

NSHapticFeedbackPerformanceTime
.Default
.Now
.DrawCompleted
Summary

Table Row Actions
Accelerator Buttons
Spring Loading
Pressure Event Stream
Alignment Feedback
Configuration
Haptic Feedback
More Information

Documentation
ForceTouchCatalog
AlignmentGuides
http://developer.apple.com/library

Technical Support
Apple Developer Forums
http://developer.apple.com/forums

General Inquiries
Paul Marcos
App Frameworks Evangelist
pmarcos@apple.com
<table>
<thead>
<tr>
<th>Related Sessions</th>
<th>Location</th>
<th>Date/Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>What’s New in Web Development in WebKit and Safari</td>
<td>Mission</td>
<td>Tuesday 9:00AM</td>
</tr>
<tr>
<td>What’s New in Cocoa</td>
<td>Presidio</td>
<td>Tuesday 1:30PM</td>
</tr>
<tr>
<td>What’s New in NSCollectionView</td>
<td>Mission</td>
<td>Thursday 4:30PM</td>
</tr>
<tr>
<td>Lab Name</td>
<td>Location</td>
<td>Date</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>--------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Cocoa and Force Touch and Gestures Lab</td>
<td>Frameworks Lab A</td>
<td>Thursday 11:00AM</td>
</tr>
<tr>
<td>Cocoa and NSCollectionView Lab</td>
<td>Frameworks Lab B</td>
<td>Friday 9:00AM</td>
</tr>
</tbody>
</table>
Apple WWDC15