

# Accessibility on OS X

New Accessibility API

Session 207

Patti Hoa

Accessibility Engineer

Chris Dolan

Accessibility Engineer

# Agenda

# Agenda

Accessibility for everyone

# Agenda

Accessibility for everyone

Overview of the new Accessibility API

# Agenda

Accessibility for everyone

Overview of the new Accessibility API

Testing of accessibility changes

# Agenda

Accessibility for everyone

Overview of the new Accessibility API

Testing of accessibility changes

Demo









Aa



Switch Control

Cursor Scaling

Mouse Keys

Speak Selection

Feature

Contrast Enhancements

Slow Keys

Sticky Keys

Invert Colors



Flash Screen Alert

Zoom

Subtitles

VoiceOver

Slow

Speak Auto-text

Closed Captioning

Zoom Window

Large Text

Best New Apps

<b>Elevate - Brain Training</b> Education Free	<b>Little Builders - Trucks, Cranes ...</b> Entertainment \$2.99	<b>Tunepics</b> Social Networking Free	<b>Appetites</b> Food & Drink Free	<b>Next - Social Music Discovery</b> Music Free	<b>BarkBuddy - Adopt a Dog Ne...</b> Lifestyle Free	<b>iMaschine</b> Music \$4.99	<b>Reverb: Effortless Personalized...</b> News Free	<b>World Soccer 2014 - Snaptee...</b> Lifestyle Free	<b>djay</b> Mus Free

--	--	--	--	--

Best New Games

<b>Ruzzle Adventure</b> Games Free	<b>Bounden</b> Games \$3.99	<b>Uncanny X-Men: Days of Future...</b> Games \$2.99	<b>F1 Race Stars™</b> Games Free	<b>OTTID</b> Games \$2.99	<b>Dragons: Rise of Berk</b> Games Free	<b>Bonsai Slice</b> Games Free	<b>GUNFINGER</b> Games Free	<b>Bill Kill'em</b> Games Free	<b>Glo of th</b> Gam \$2.99

--	--	--	--	--

Backyard BBQ

--	--	--	--	--	--	--	--	--

iPhone

All Categories

**QUICK LINKS**

- About In-App Purchases
- Parents' Guide to iTunes
- About Kids Apps & Games
- New to the App Store?
- Editors' Choice
- App Collections
- Game Collections
- Apps Made by Apple
- Redeem
- Purchased
- App Store on Facebook
- App Store on Twitter
- Send iTunes Gifts

TOP PAID APPS >

- Blek Games**
- Heads Up!**  
Games
- A Dark Room**  
Games
- Minecraft - Pocket Edition**  
Games
- Facetune**  
Photo & Video
- Afterlight**  
Photo & Video
- Geometry Dash**  
Games
- Plague Inc.**  
Games
- LOVE by Tiny Buddha**  
Lifestyle

# Flow of Accessibility Information



Application

# Flow of Accessibility Information



Assistive Software



Application

# Flow of Accessibility Information



Request accessibility information

# Flow of Accessibility Information

Vends accessibility information



Assistive Software



Application

Request accessibility information





# Flow of Accessibility Information

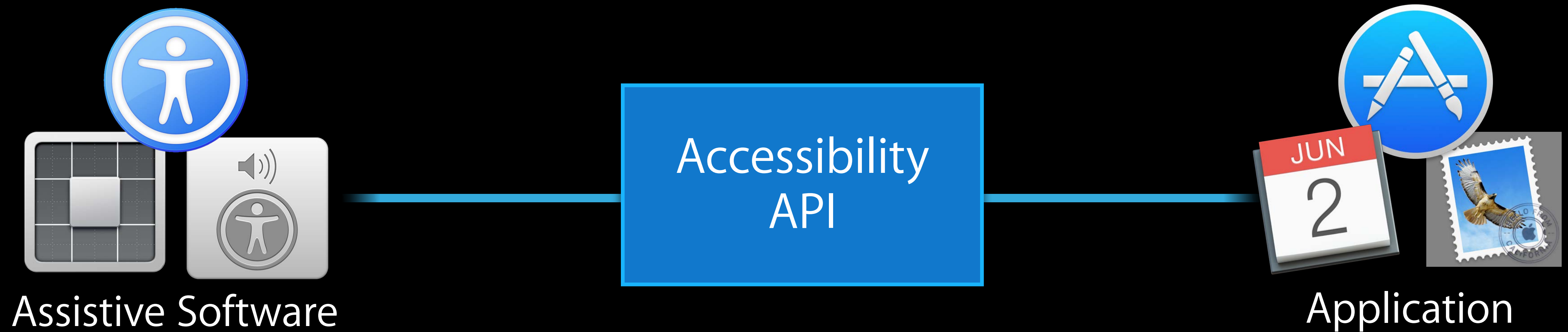


Assistive Software



Application

# Flow of Accessibility Information





# New Accessibility API on OS X

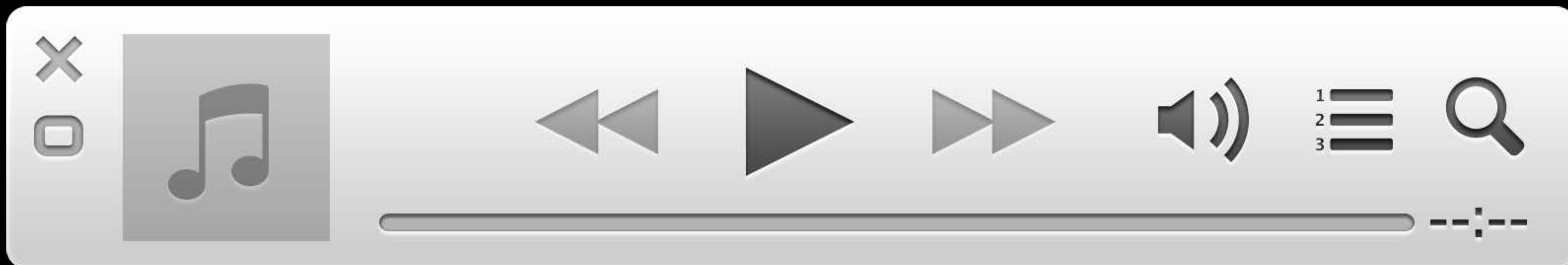


Goals:

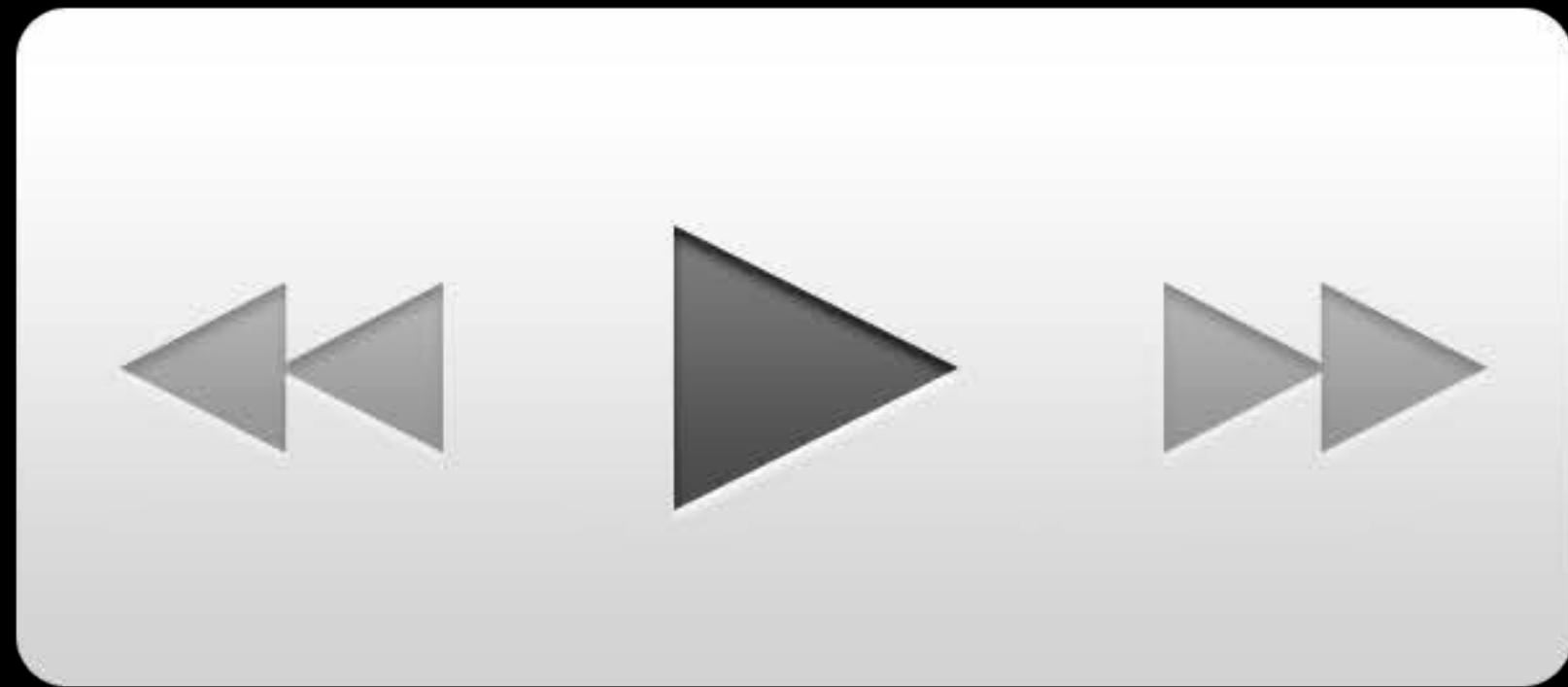
Goal:

Goal:

Simplify

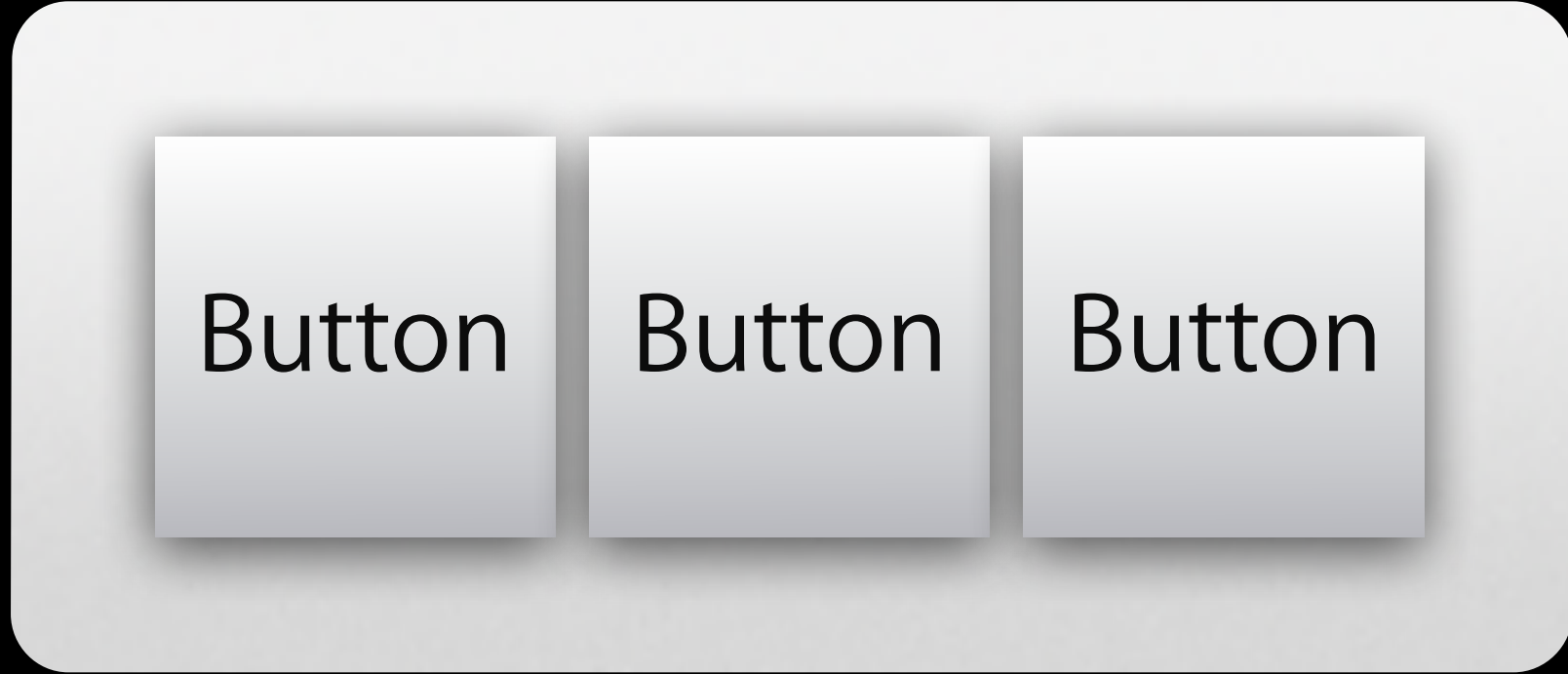


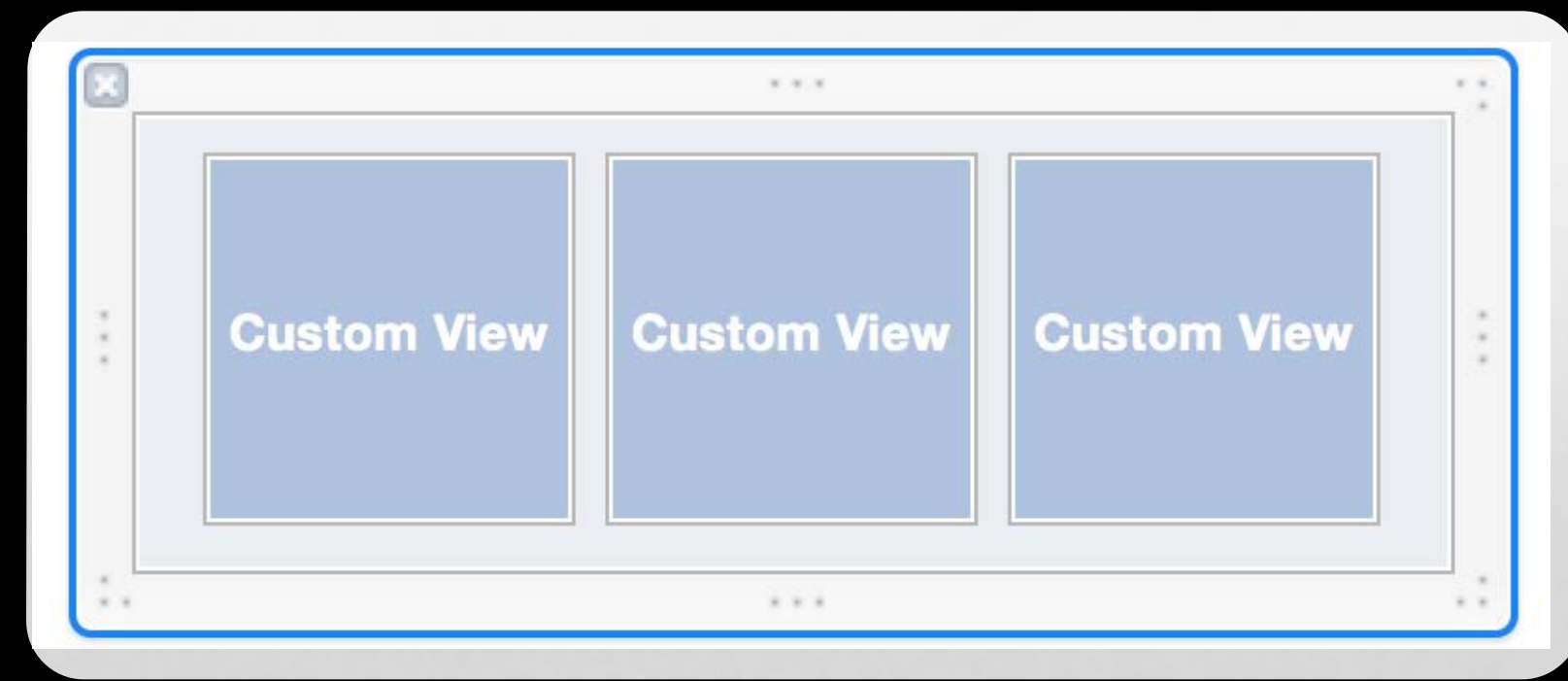


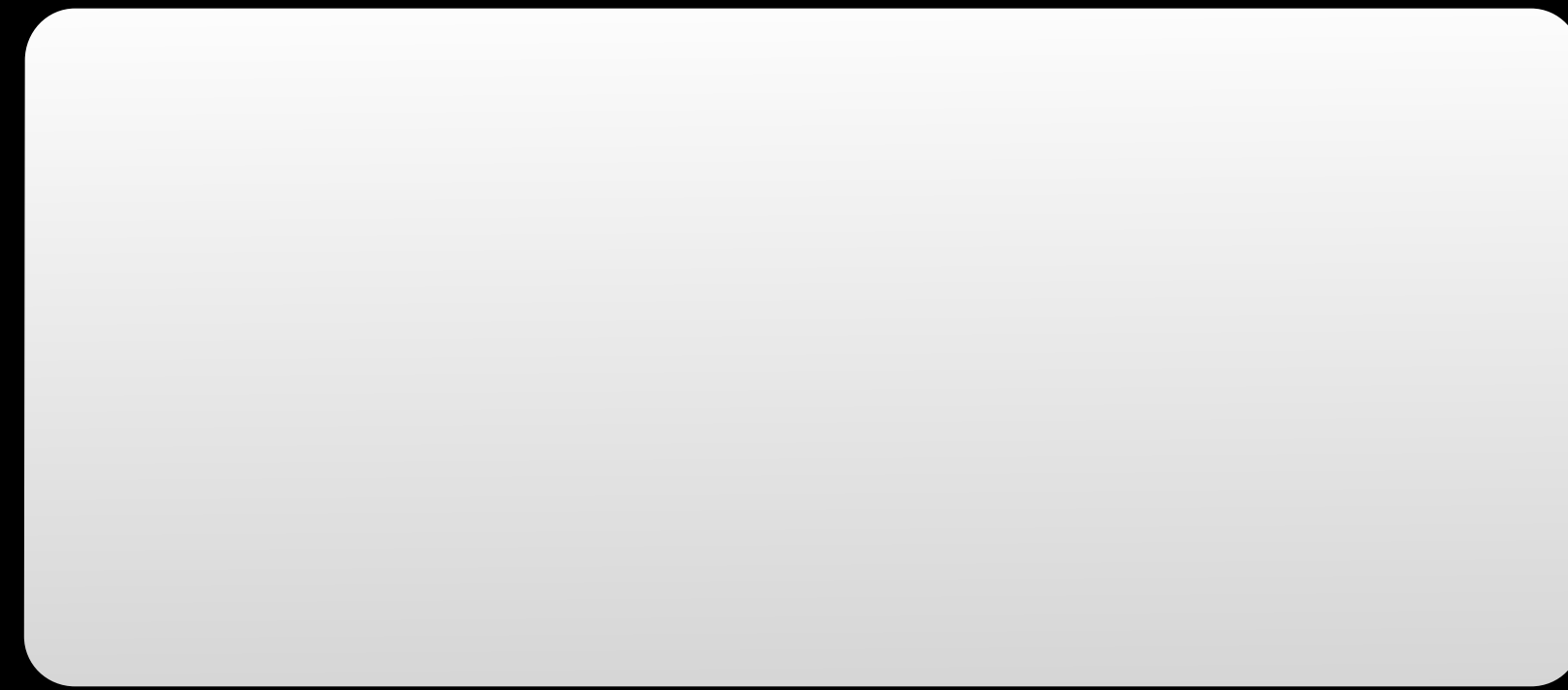


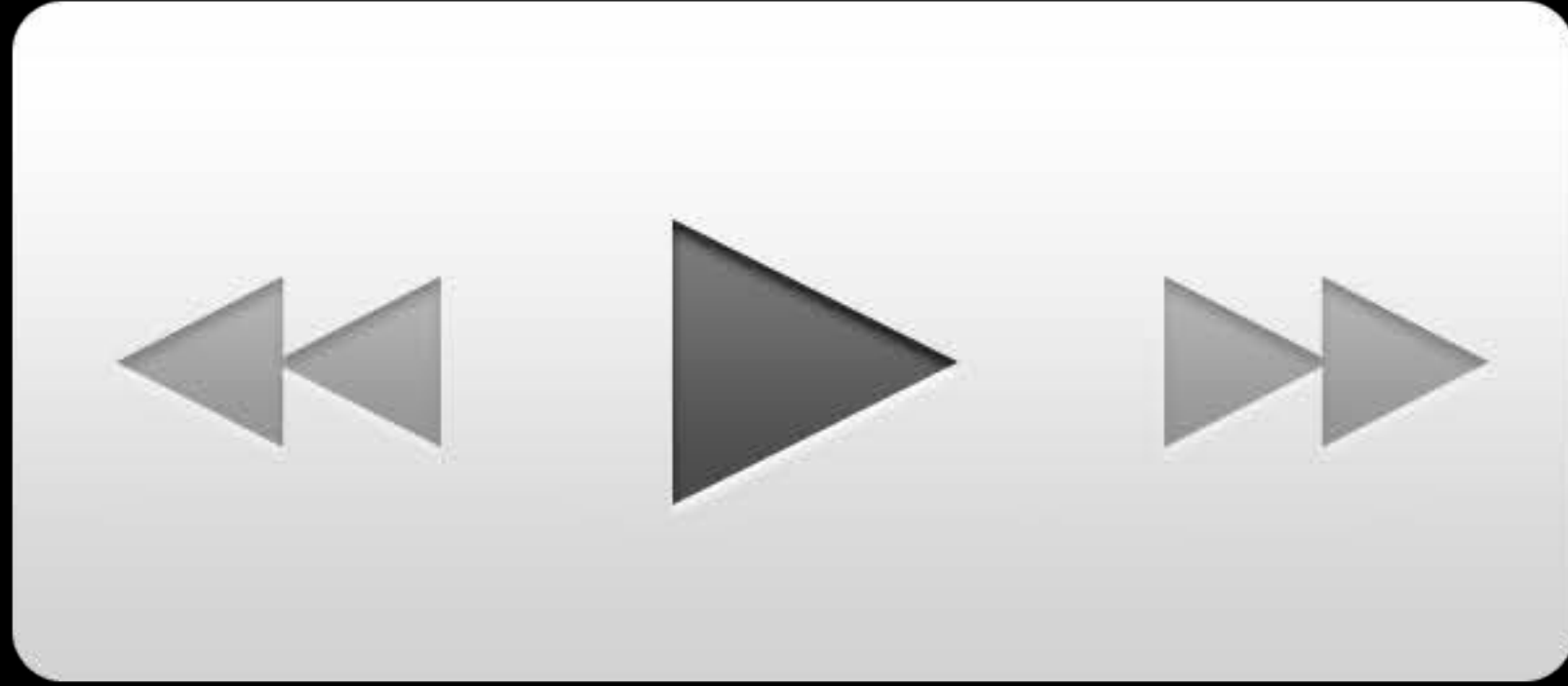
















Using AppKit control?

Using AppKit control?

**Accessibility for free!**

button



**Custom Class**

Class

**Identity**

Identifier

**Tool Tip**

Tool Tip

**Accessibility Identity**

Description

Help

**User Defined Runtime Attributes**

Key Path	Type	Value
----------	------	-------

Custom Class

Class

Identity

Identifier

Tool Tip

Tool Tip

Accessibility Identity

Description

Help

User Defined Runtime Attributes

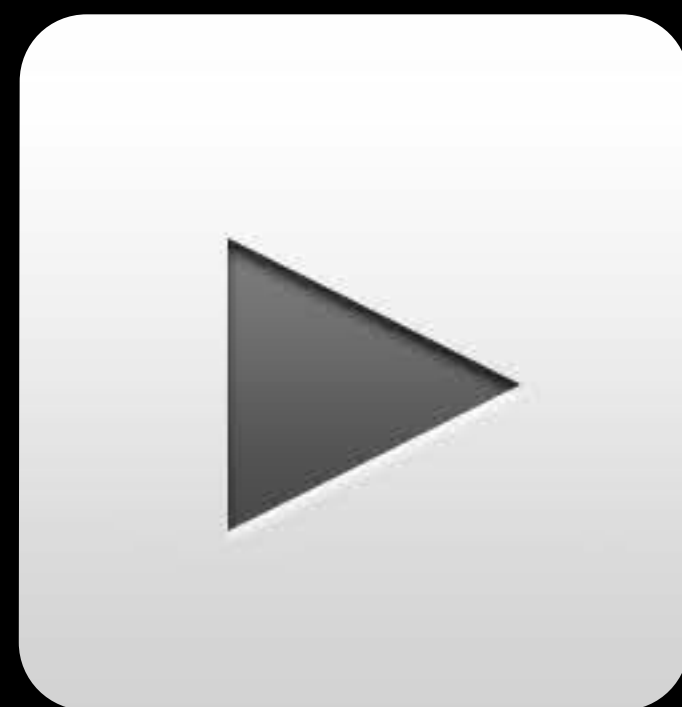
Key Path	Type	Value
----------	------	-------



# Adding Label to an AppKit Control

```
- (void)awakeFromNib {  
    .....  
    NSButtonCell *myButtonCell = [myLoopButton cell];  
    myButtonCell.accessibilityLabel = NSLocalizedString(@"play", nil);  
    .....  
}
```

play button



# Accessibility Properties



# Accessibility Properties

Majority of attributes are now properties

```
@property (copy) NSString *accessibilityLabel  
@property NSRect accessibilityFrame
```

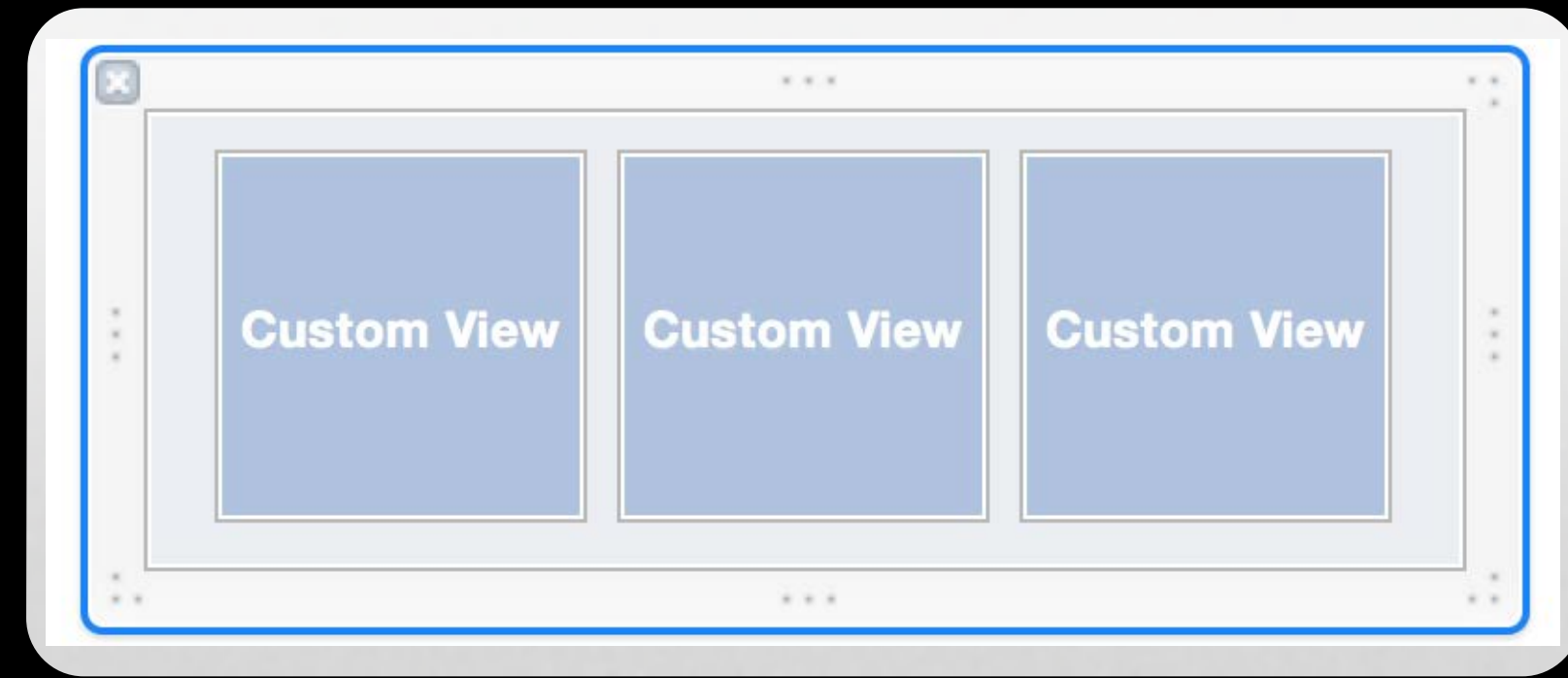
# Accessibility Properties

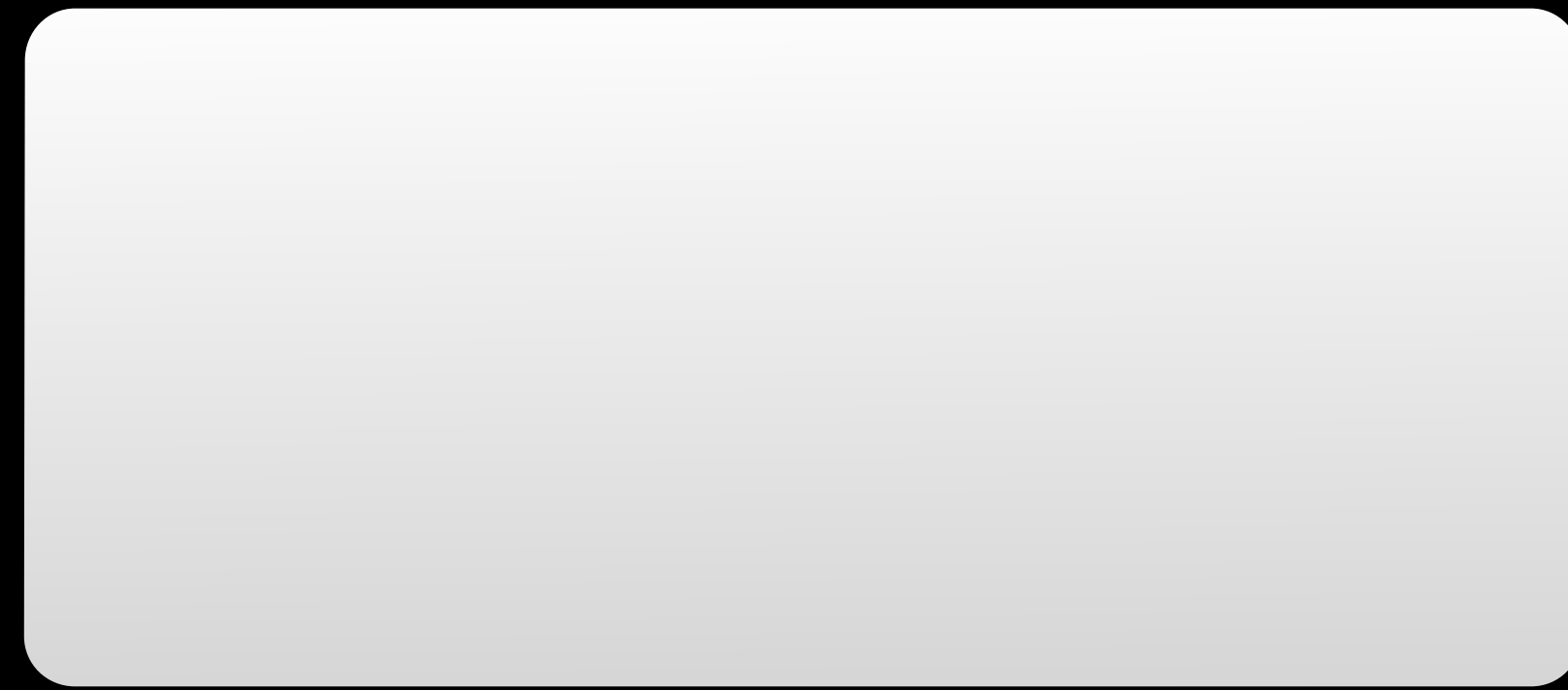
Majority of attributes are now properties

```
@property (copy) NSString *accessibilityLabel  
@property NSRect accessibilityFrame
```

Subclasses no longer needed

```
searchField.accessibilityLabel = NSLocalizedString(@"play", nil);  
searchField.accessibilityLinkedUIElements = @[content];
```





# Accessibility Protocols

Guide developers through implementation

Covers most common types of UI

# Pick a Protocol

Button

CheckBox

Group

Image

StaticText

Stepper

...

# Implement the Protocol

```
@interface MyButtonView : NSView <NSAccessibilityButton>
@end
```

# Implement the Protocol

```
@interface MyButtonView : NSView <NSAccessibilityButton>
@end
```

```
10
11 @implementation MyButtonView
12     ⚠ Method 'accessibilityPerformPress' in protocol 'NSAccessibilityButton' not implemented
13     ⚠ Method 'accessibilityLabel' in protocol 'NSAccessibilityButton' not implemented
14
```



# Implement the Protocol

# Implement the Protocol

```
@implementation MyButtonView
- (NSString *)accessibilityLabel {
    return NSLocalizedStringForKey(@"play", nil);
}
- (BOOL)accessibilityPerformPress {
    [self press];
    return YES;
}
@end
```

# Implement the Protocol

```
@implementation MyButtonView
- (NSString *)accessibilityLabel {
    return NSLocalizedStringForKey(@"play", nil);
}
- (BOOL)accessibilityPerformPress {
    [self press];
    return YES;
}
@end
```

That is all!

AppKit Infers More

# AppKit Infers More

Infer isAccessibilityElement

# AppKit Infers More

Infer `isAccessibilityElement`

Auto-populate some attributes

# Protocols

# Protocols

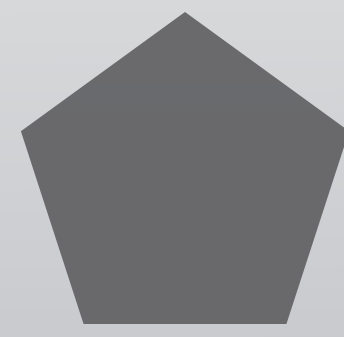
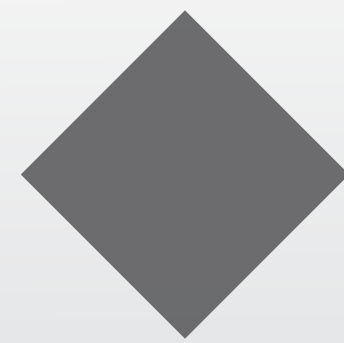
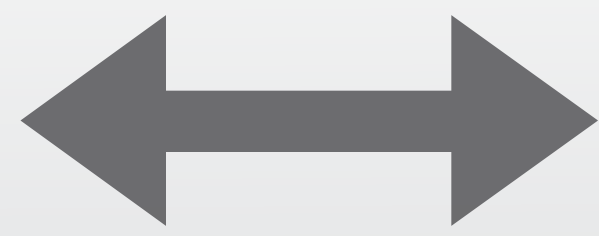
```
@protocol NSAccessibilityElement <NSObject>
@required
- (CGRect)accessibilityFrame;
- (id)accessibilityParent;
@end
```

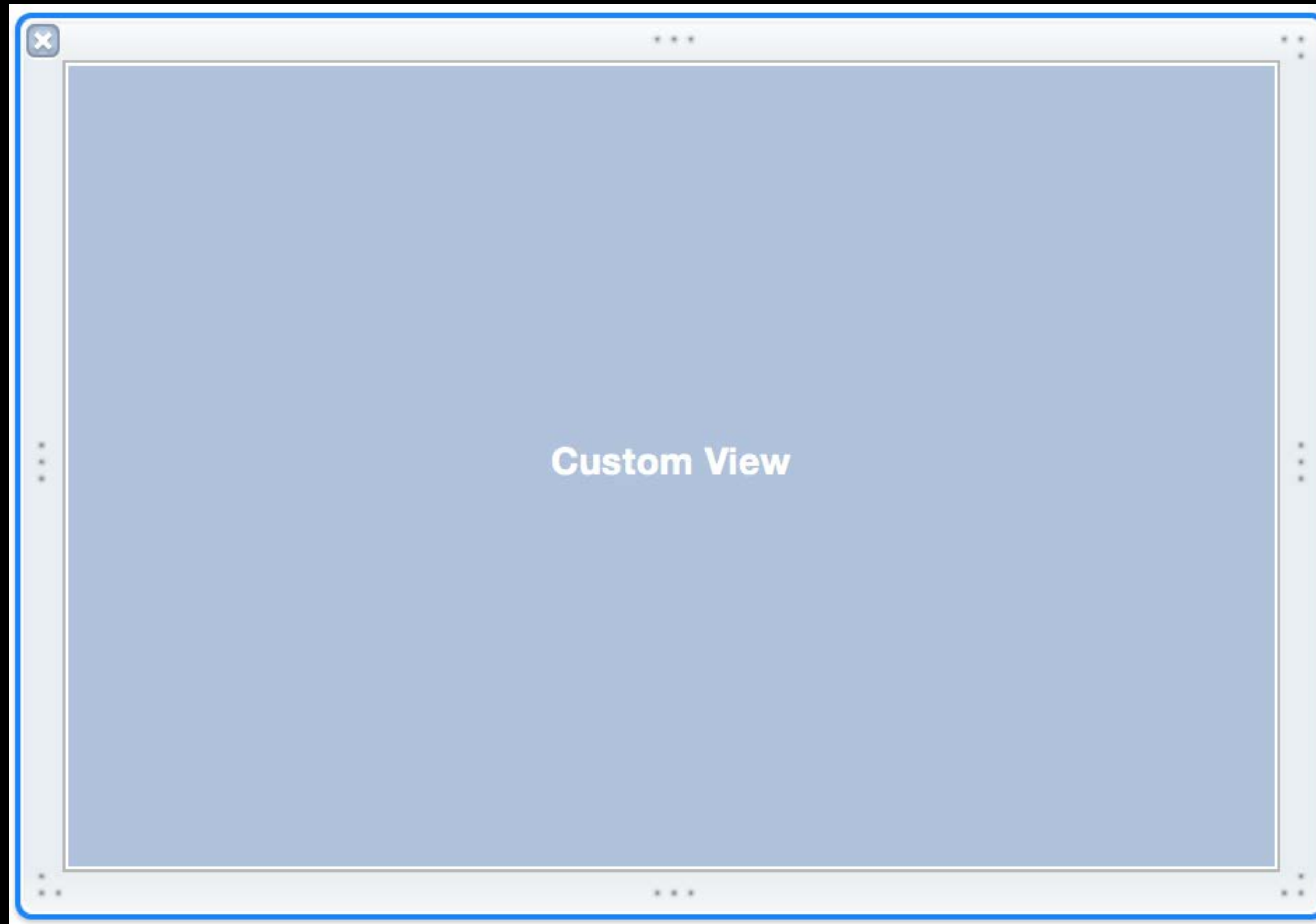


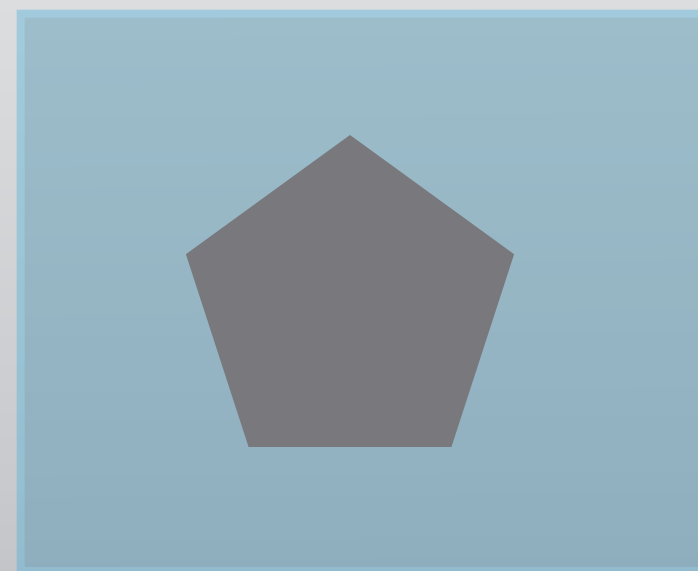
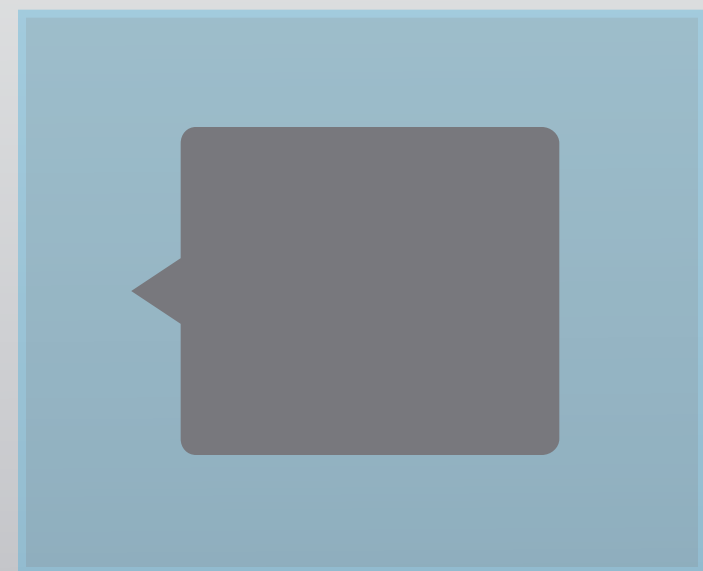
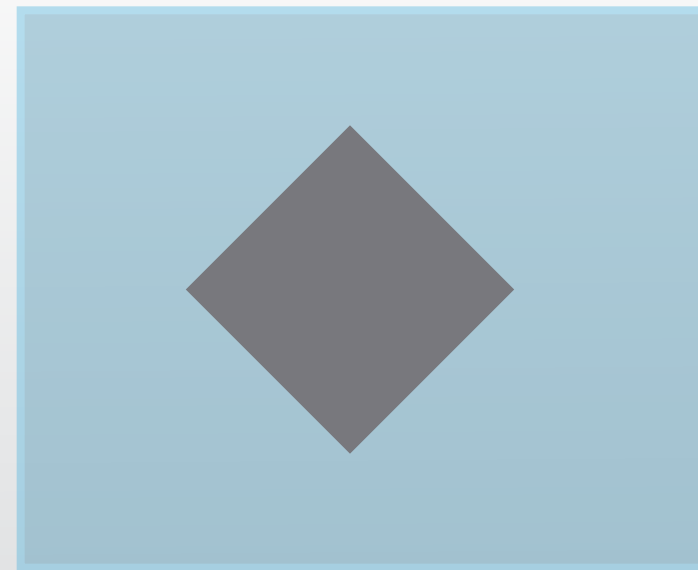
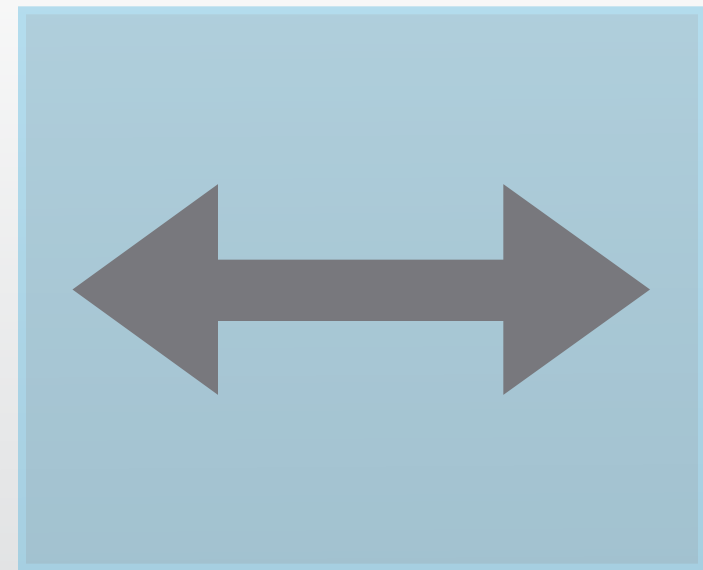
# Protocols

```
@protocol NSAccessibilityElement <NSObject>
@required
- (CGRect)accessibilityFrame;
- (id)accessibilityParent;
@end
```

```
@protocol NSAccessibilityButton <NSAccessibilityElement>
@required
- (NSString *)accessibilityLabel;
- (BOOL)accessibilityPerformPress;
@end
```







# NSAccessibilityElement

UI without backing NSView

# NSAccessibilityElement

```
- (NSArray *)accessibilityChildren {
    if ( _accessibilityChildren == nil ) {
        ...

        NSAccessibilityElement *imageElement = [NSAccessibilityElement new];
        NSRect bounds = NSAccessibilityFrameInView(self, [self imageBounds]);

        [imageElement setAccessibilityParent:self];
        [imageElement setAccessibilityRole:NSAccessibilityImageRole];
        [imageElement setAccessibilityLabel:accessibilityLabel];
        [imageElement setAccessibilityFrame:bounds];

        [_accessibilityChildren addObject:imageElement];
    }
    return _accessibilityChildren;
}
```

# NSAccessibilityElement

```
- (NSArray *)accessibilityChildren {  
    if ( _accessibilityChildren == nil ) {
```

```
        ...
```

```
        NSAccessibilityElement *imageElement = [NSAccessibilityElement new];  
        NSRect bounds = NSAccessibilityFrameInView(self, [self imageBounds]);
```

```
        [imageElement setAccessibilityParent:self];
```

```
        [imageElement setAccessibilityRole:NSAccessibilityImageRole];
```

```
        [imageElement setAccessibilityLabel:accessibilityLabel];
```

```
        [imageElement setAccessibilityFrame:bounds];
```

```
        [_accessibilityChildren addObject:imageElement];
```

```
    }
```

```
    return _accessibilityChildren;
```

```
}
```

# NSAccessibilityElement

```
- (NSArray *)accessibilityChildren {
    if ( _accessibilityChildren == nil ) {
        ...

        NSAccessibilityElement *imageElement = [NSAccessibilityElement new];
        NSRect bounds = NSAccessibilityFrameInView(self, [self imageBounds]);

        [imageElement setAccessibilityParent:self];
        [imageElement setAccessibilityRole:NSAccessibilityImageRole];
        [imageElement setAccessibilityLabel:accessibilityLabel];
        [imageElement setAccessibilityFrame:bounds];

        [_accessibilityChildren addObject:imageElement];
    }
    return _accessibilityChildren;
}
```



# NSAccessibilityElement

```
- (NSArray *)accessibilityChildren {  
    if ( _accessibilityChildren == nil ) {  
        ...
```

```
        NSAccessibilityElement *imageElement = [NSAccessibilityElement new];  
        NSRect bounds = NSAccessibilityFrameInView(self, [self imageBounds]);
```

```
        [imageElement setAccessibilityParent:self];  
        [imageElement setAccessibilityRole:NSAccessibilityImageRole];  
        [imageElement setAccessibilityLabel:accessibilityLabel];  
        [imageElement setAccessibilityFrame:bounds];
```

```
        [_accessibilityChildren addObject:imageElement];
```

```
    }
```

```
    return _accessibilityChildren;
```

```
}
```

# NSAccessibilityElement

```
- (NSArray *)accessibilityChildren {
    if ( _accessibilityChildren == nil ) {
        ...

        NSAccessibilityElement *imageElement = [NSAccessibilityElement new];
        NSRect bounds = NSAccessibilityFrameInView(self, [self imageBounds]);

        [imageElement setAccessibilityParent:self];
        [imageElement setAccessibilityRole:NSAccessibilityImageRole];
        [imageElement setAccessibilityLabel:accessibilityLabel];
        [imageElement setAccessibilityFrame:bounds];

        [_accessibilityChildren addObject:imageElement];
    }
    return _accessibilityChildren;
}
```

Already implemented accessibility?

Already implemented accessibility?

**No extra work!**

Binary and source compatible

# Agenda

Accessibility for everyone

Overview of the new Accessibility API

Testing



Accessibility Inspector







Accessibility Inspector (Locked)

▼ Hierarchy

- ▼ AXApplication
  - ▼ AXWindow:AXStandardWindow
    - AXButton

▼ Attributes

isAccessibilityEnabled	NO
accessibilityLabel	play
accessibilityParent	<AXWindow:AXStandardWindow>
isAccessibilityFocused	NO
accessibilityRole	AXButton
accessibilityTopLevelUIElement	<AXWindow:AXStandardWindow>
accessibilityTitle	<nil>
accessibilityWindow	<AXWindow:AXStandardWindow>
accessibilityRoleDescription	button
accessibilitySubrole	<nil>
accessibilityFrame	x=65.00 y=85.00 w=30.00 h=30.00

▼ Actions

- accessibilityPerformPress

No Selection

⏏ ⌘F7 toggles element lock

▲ ▼ ◀ ▶ ↻ □



VoiceOver



Finder File Edit View Go Window Help [System Status Icons] Mon 9:41 AM

Apple - Accessibility

www.apple.com/accessibility/


Store Mac iPod iPhone iPad iTunes Support

# Accessibility

OS X iOS Third-Party Resources

## We've done everything possible to make anything possible.

Apple products are simple, intuitive, and easy to use. And to help you do more in more ways, a variety of award-winning assistive technologies come standard. So every device not only has accessible features — but accessible principles — built right in.

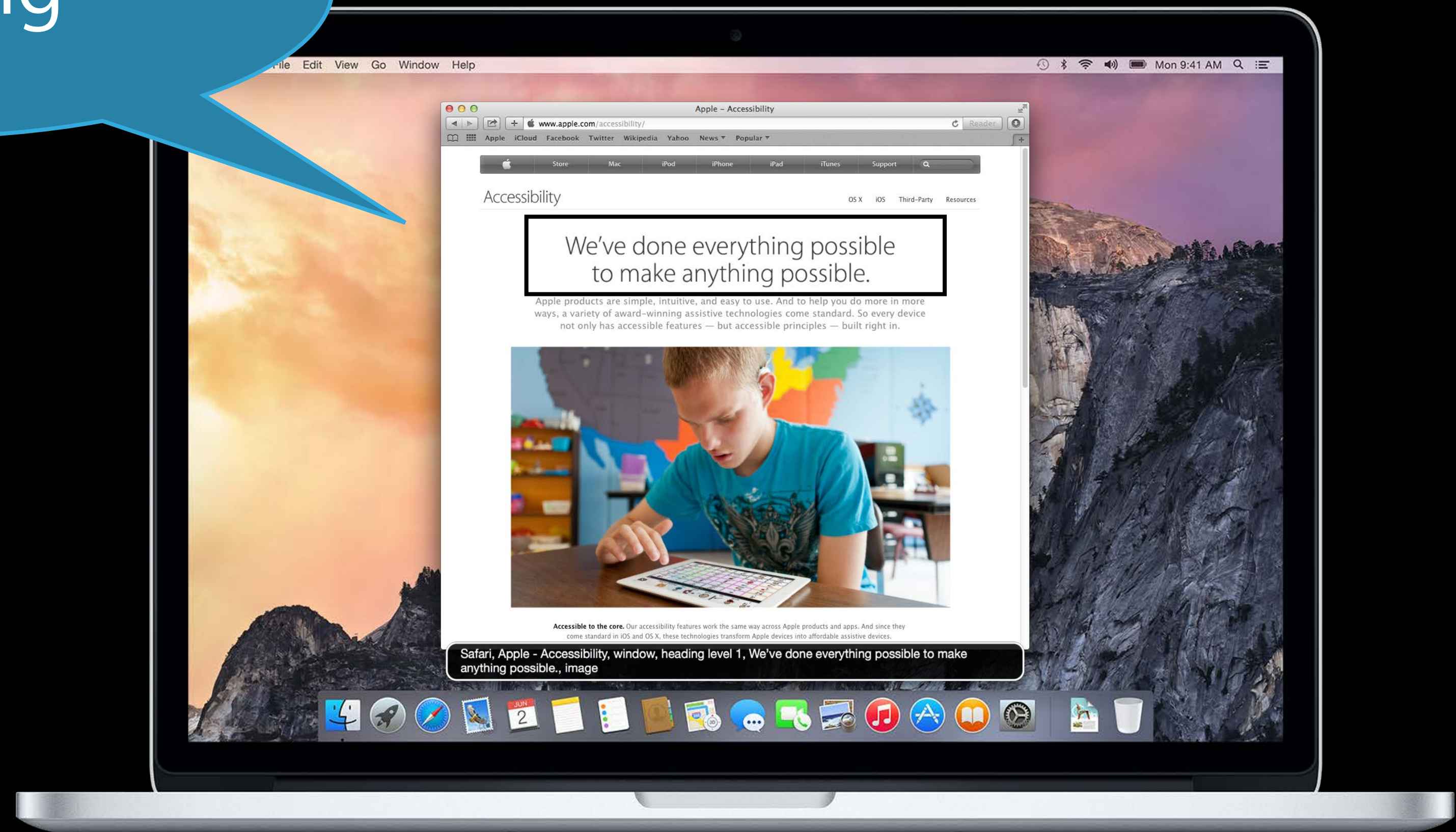


**Accessible to the core.** Our accessibility features work the same way across Apple products and apps. And since they come standard in iOS and OS X, these technologies transform Apple devices into affordable assistive devices.

Safari, Apple - Accessibility, window, heading level 1, We've done everything possible to make anything possible., image



“Heading”





Accessibility

< > Show All Search

Vision

- Display
- Zoom
- VoiceOver**

Media

- Descriptions
- Captions

Hearing

- Audio

Interacting

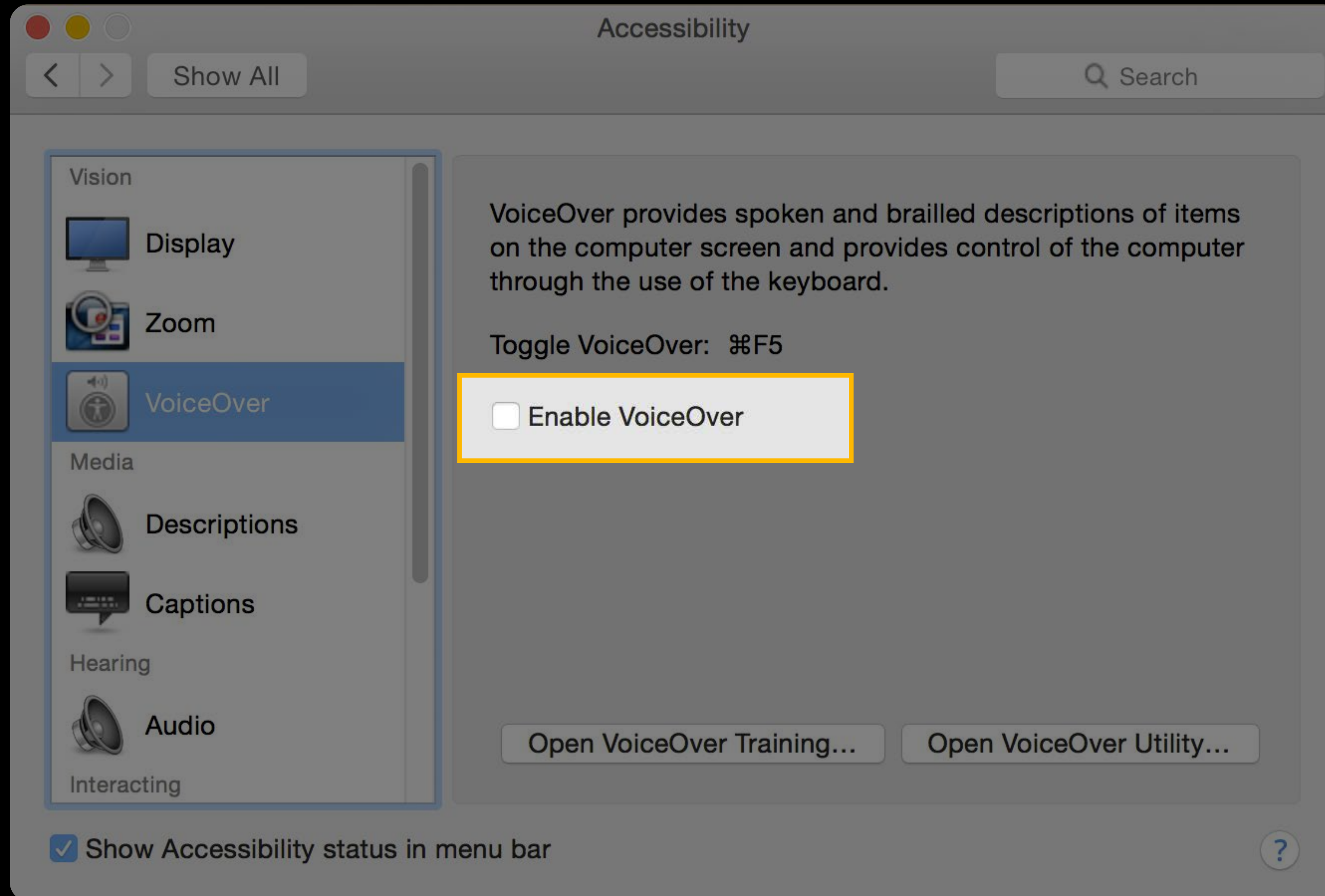
VoiceOver provides spoken and brailled descriptions of items on the computer screen and provides control of the computer through the use of the keyboard.

Toggle VoiceOver: ⌘F5

Enable VoiceOver

Open VoiceOver Training... Open VoiceOver Utility...

Show Accessibility status in menu bar ?





VoiceOver



## Welcome to VoiceOver

VoiceOver speaks descriptions of items on the screen and can be used to control the computer using only your keyboard.

Do not show this message again

Learn More

Use VoiceOver

Turn Off VoiceOver

# VoiceOver Commands

# VoiceOver Commands

Next Item:



# VoiceOver Commands

Next Item:



Previous Item:



# VoiceOver Commands

Next Item:



Previous Item:



Activate Item:



# VoiceOver Commands

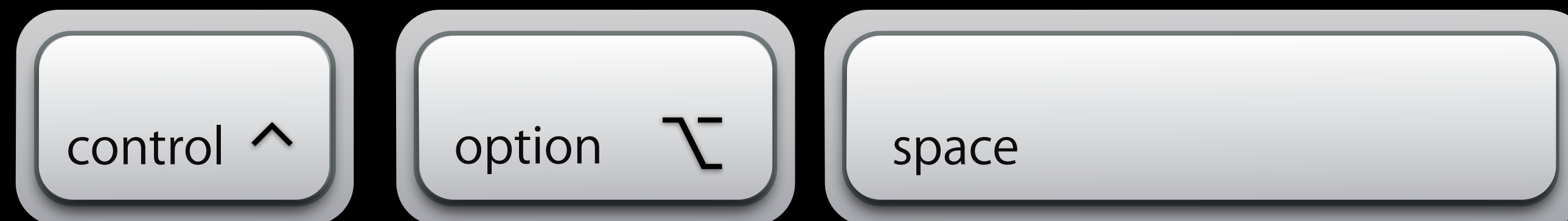
Next Item:



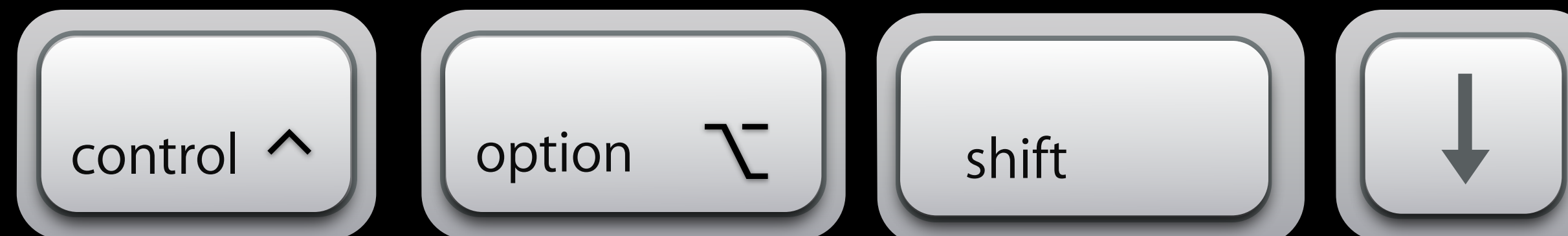
Previous Item:



Activate Item:



Enter Group:



# VoiceOver Commands

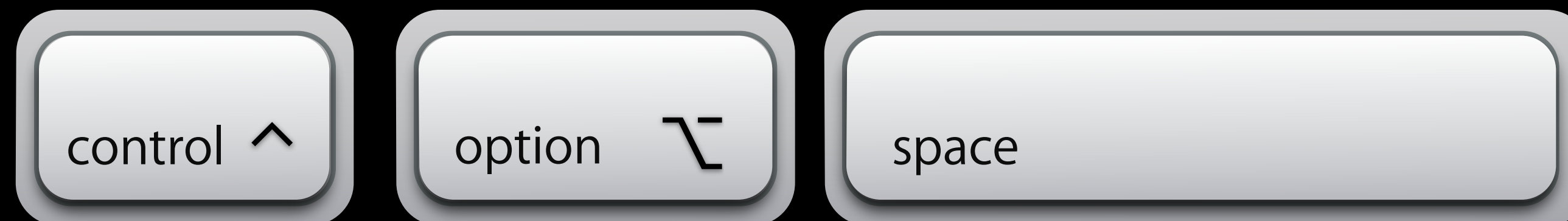
Next Item:



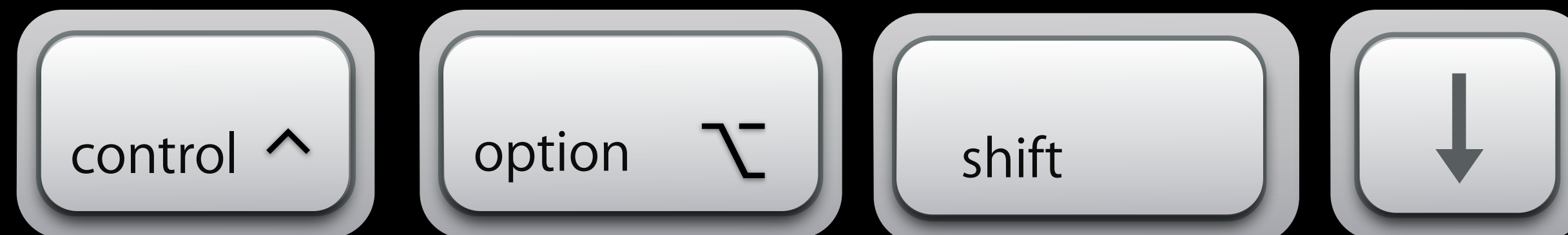
Previous Item:



Activate Item:



Enter Group:



Leave Group:



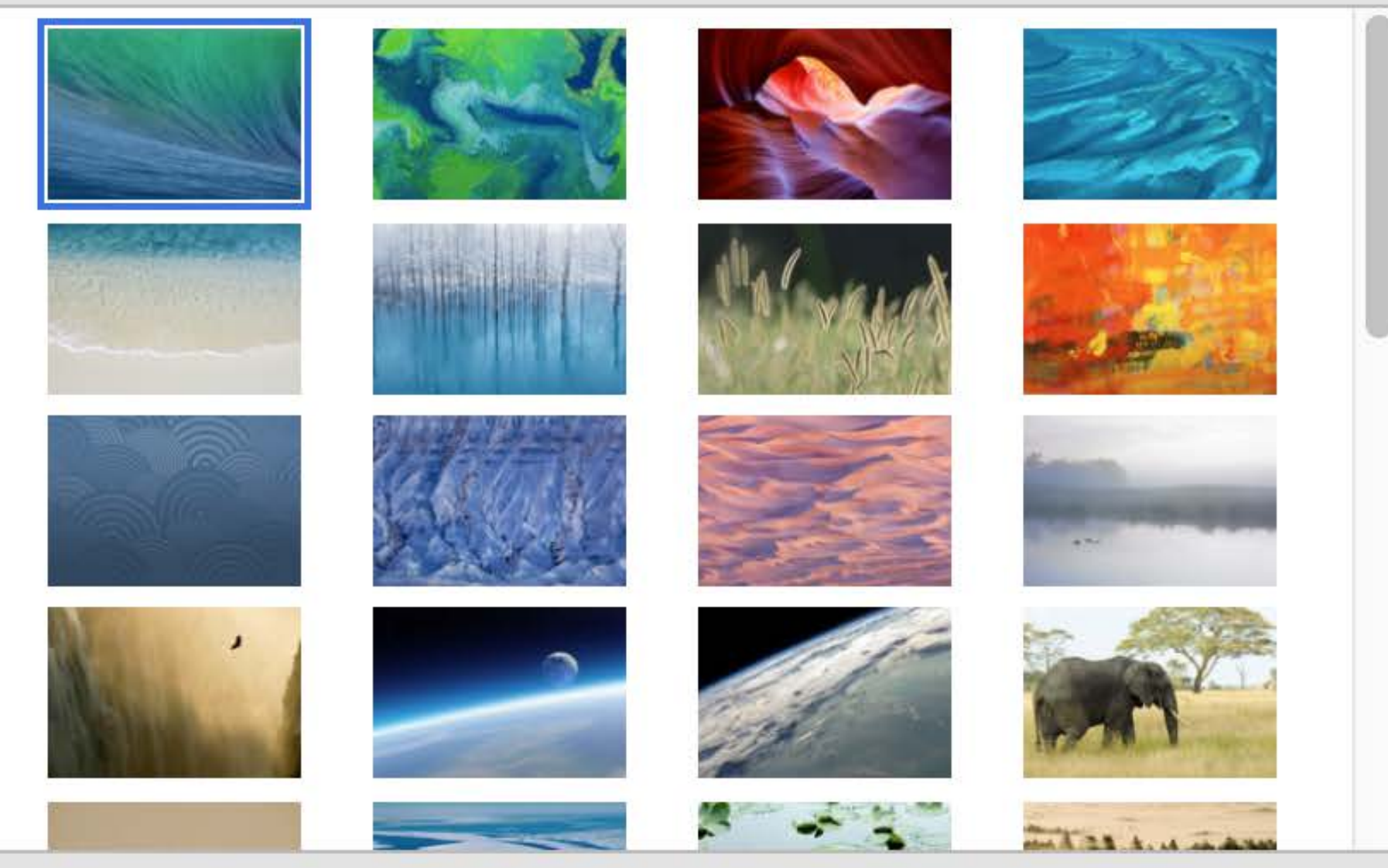
Desktop & Screen Saver

< > Show All Search

Desktop Screen Saver

Wave

▼ Apple  
Desktop Pictures  
Solid Colors  
▼ Folders  
Pictures



+ -

Change picture: Every 30 minutes

Random order

Translucent menu bar

?



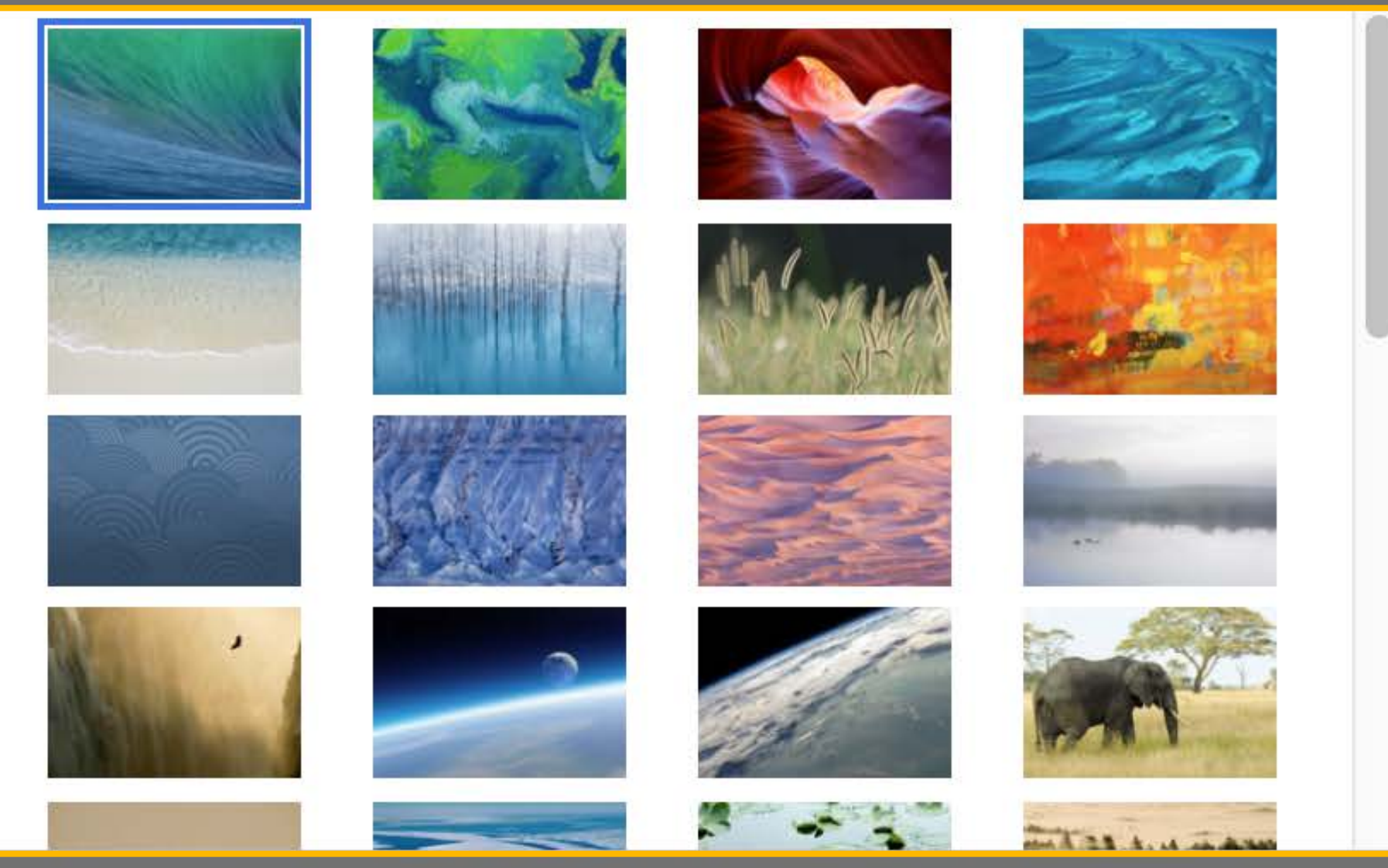
Desktop & Screen Saver

< > Show All Search

Desktop Screen Saver

Wave

▼ Apple  
Desktop Pictures  
Solid Colors  
▼ Folders  
Pictures



+ -

Change picture: Every 30 minutes

Random order

Translucent menu bar

?

# VoiceOver Commands

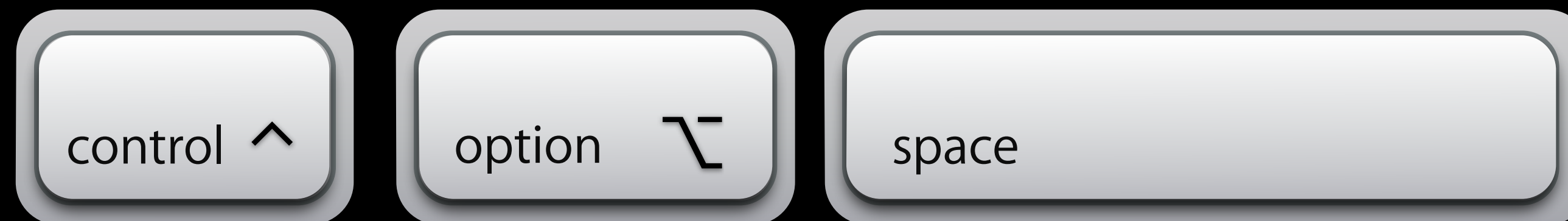
Next Item:



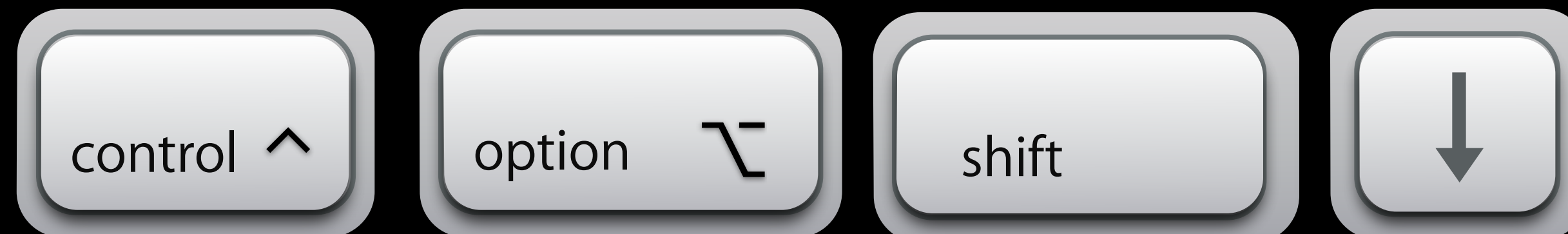
Previous Item:



Activate Item:



Enter Group:



Leave Group:

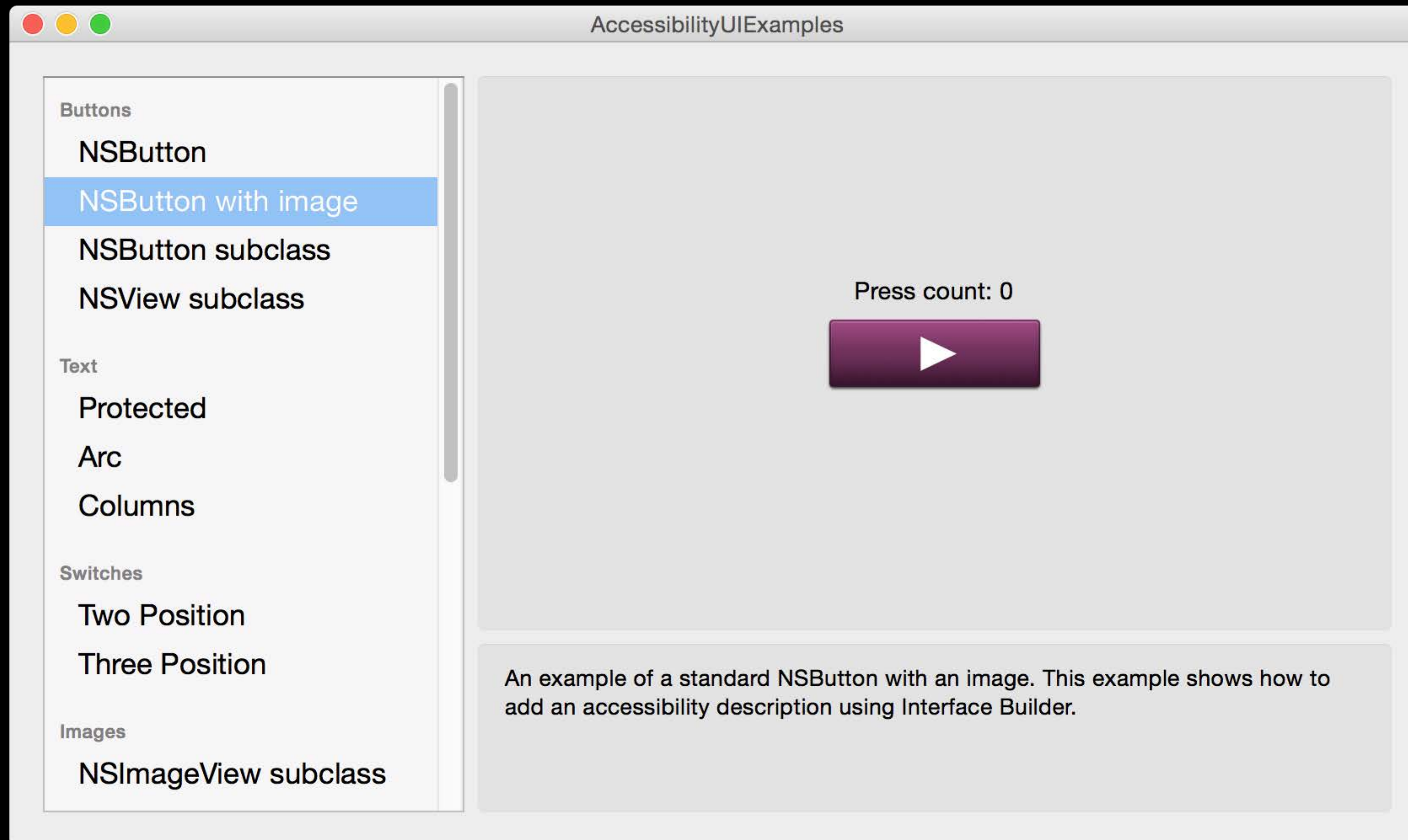


*Demo*

Tic Tac Toe

Chris Dolan

Accessibility Engineer



<http://developer.apple.com>

# Summary

Brand new, easy to use API

Test with VoiceOver, Accessibility Inspector

Developer examples

# More Information

Jake Behrens

App Frameworks Evangelist

[behrens@apple.com](mailto:behrens@apple.com)

Documentation

<http://developer.apple.com>

Apple Developer Forums

<http://devforums.apple.com>

# Related Sessions

- 
- Accessibility on iOS Russian Hill Tuesday 3:15PM
  - Improving the Accessibility and Usability of Complex Web Applications Marina Friday 9:00AM
-

# Labs

- 
- Accessibility and Speech Lab Frameworks Lab B Wednesday 10:15AM
-



 WWDC14