Frameworks

#WWDC14

Advanced Topics in Internationalization

Session 201

Doug Davidson
Software Engineer

Deborah Goldsmith
Software Engineer

Karan Misra
Software Engineer

Introduction

Outline

What's new in internationalization

Languages and locales

Case studies

- Formatting dates and names
- Right-to-left and bidirectional text
- Keyboard traits, sizing, and positioning
- Marked text

What’s New in Internationalization?

What's New in Internationalization?

New localizations and keyboards

What’s New in Internationalization?

New localizations and keyboards

Language and region settings

What’s New in Internationalization?

New localizations and keyboards

Language and region settings

Lunar calendar support

What's New in Internationalization?

New localizations and keyboards

Language and region settings

Lunar calendar support

String encoding detection

What's New in Internationalization?

New localizations and keyboards

Language and region settings

Lunar calendar support

String encoding detection

New formatters

What's New in Internationalization?

New localizations and keyboards

Language and region settings

Lunar calendar support

String encoding detection

New formatters

Formatting context support

New iOS Localizations and Keyboards

New iOS Localizations and Keyboards

Localizations

- Hindi
- Indian English
- Canadian French
- Hong Kong Chinese

New iOS Localizations and Keyboards

Localizations

- Hindi
- Indian English
- Canadian French
- Hong Kong Chinese

Keyboards

- Bengali
- Marathi
- Urdu
- Indian English
- Filipino
- Slovenian

Language and Region Settings

Language and Region Settings

Specify primary language

Language and Region Settings

Specify primary language

Also specify preferred languages in order

Language and Region Settings

Specify primary language

Also specify preferred languages in order

Allows localization into other languages

Language and Region Settings

Specify primary language

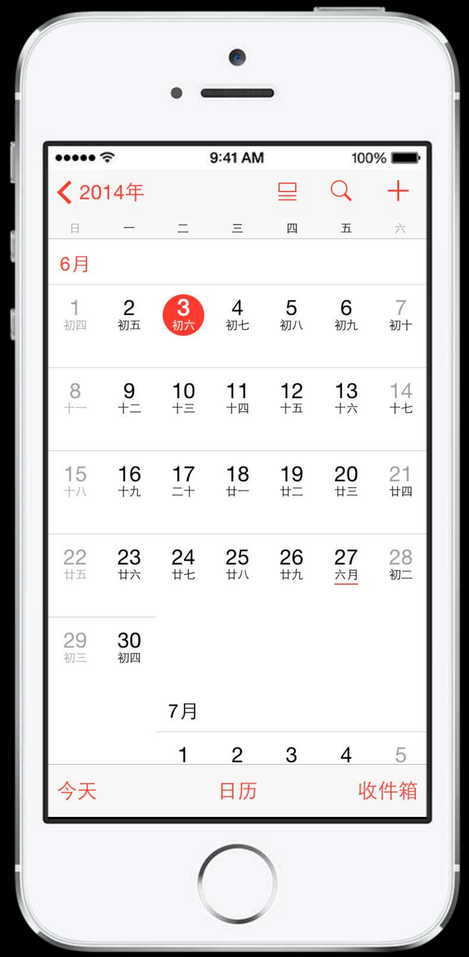
Also specify preferred languages in order

Allows localization into other languages

Can specify region format language independently

Lunar Calendar Support

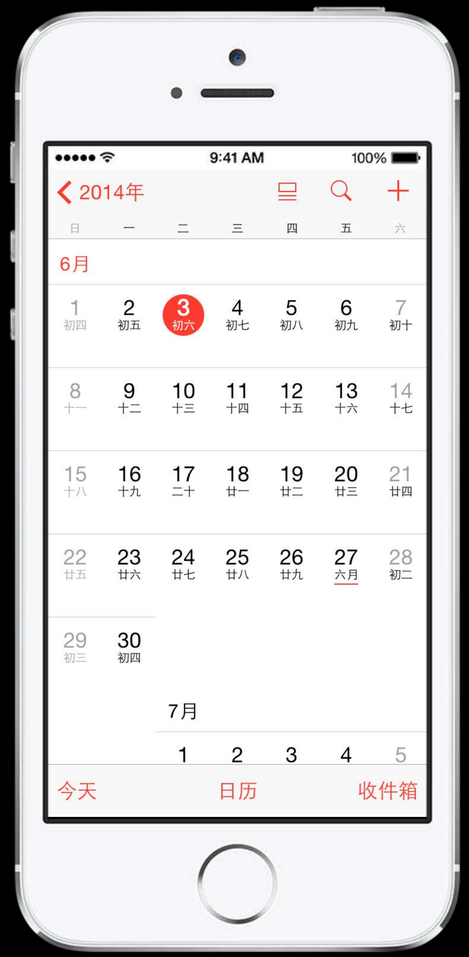
NEW



Lunar Calendar Support

NEW

Lunar calendar display

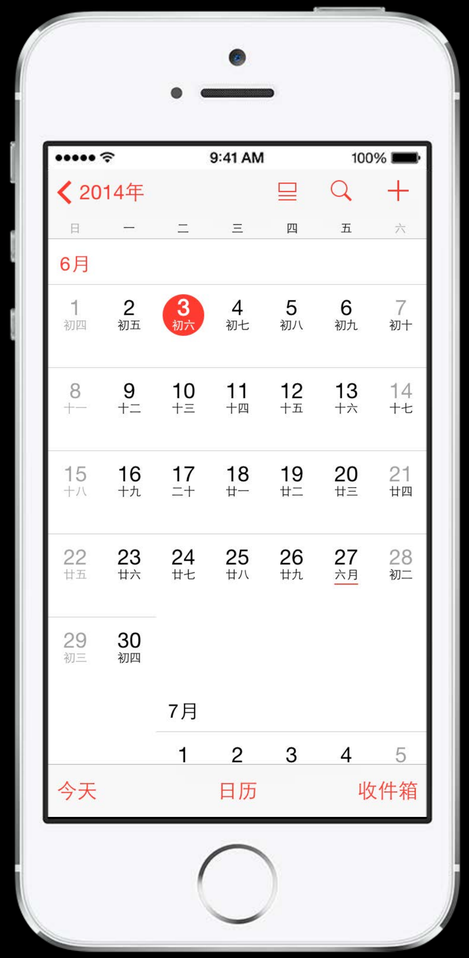


Lunar Calendar Support

NEW

Lunar calendar display

Two new Islamic calendars



Lunar Calendar Support

NEW

Lunar calendar display

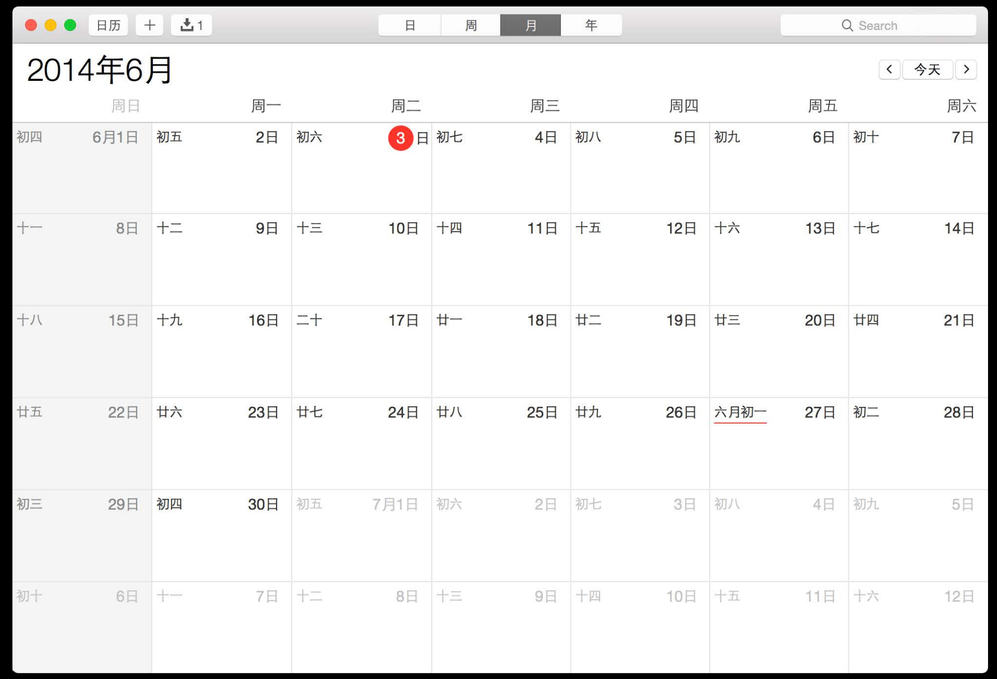
Two new Islamic calendars

Lunar Calendar Support

NEW

Lunar calendar display

Two new Islamic calendars

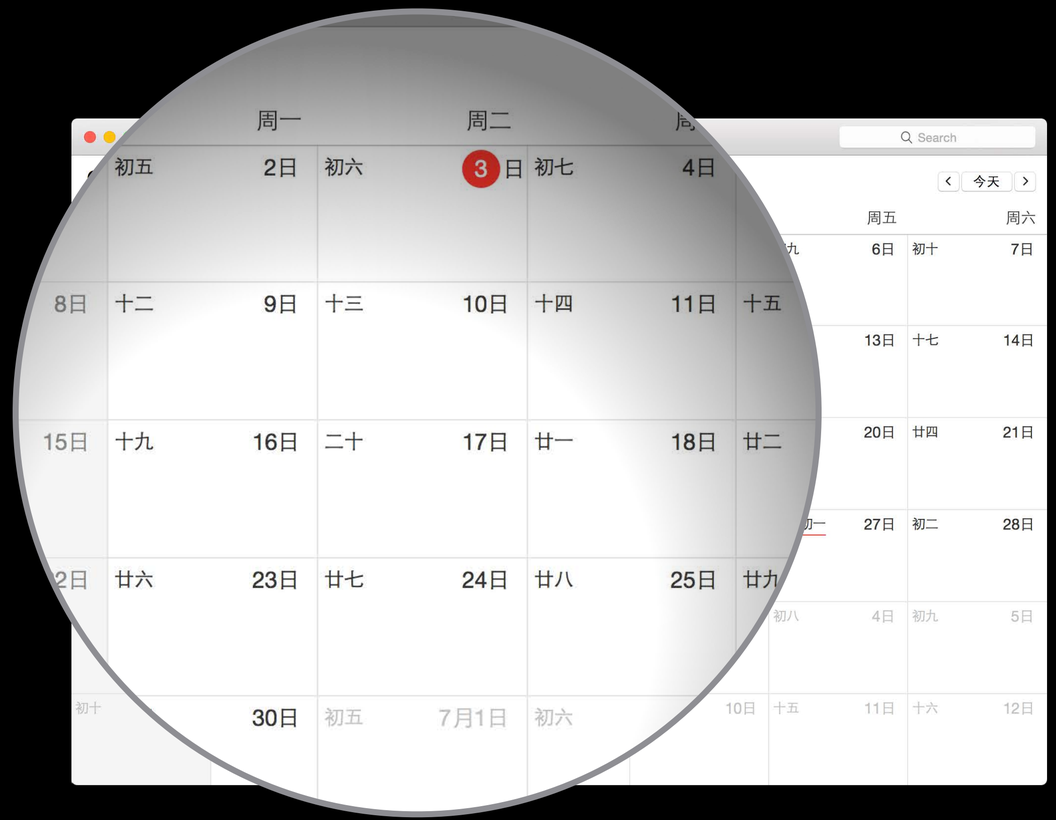


Lunar Calendar Support

NEW

Lunar calendar display

Two new Islamic calendars



String Encoding Detection

NEW

```
+ (NSStringEncoding)stringEncodingForData:(NSData *)data
                          encodingOptions:(NSDictionary *)opts
                          convertedString:(NSString **)string
                      usedLossyConversion:(BOOL *)usedLossyConversion;
```

String Encoding Detection


```
+ (NSStringEncoding)stringEncodingForData:(NSData *)data
                          encodingOptions:(NSDictionary *)opts
                          convertedString:(NSString **)string
                      usedLossyConversion:(BOOL *)usedLossyConversion;
```

Detection, conversion, or both

String Encoding Detection


```
+ (NSStringEncoding)stringEncodingForData:(NSData *)data
                          encodingOptions:(NSDictionary *)opts
                          convertedString:(NSString **)string
                      usedLossyConversion:(BOOL *)usedLossyConversion;
```

Detection, conversion, or both

Lossy or non-lossy conversion

String Encoding Detection


```
+ (NSStringEncoding)stringEncodingForData:(NSData *)data
                          encodingOptions:(NSDictionary *)opts
                          convertedString:(NSString **)string
                      usedLossyConversion:(BOOL *)usedLossyConversion;
```

Detection, conversion, or both

Lossy or non-lossy conversion

Can specify likely or required encodings

NSDateComponentsFormatter

NSDateComponentsFormatter

For display of amounts or durations of time (e.g., “1 hour, 50 minutes”)

NSDateComponentsFormatter

For display of amounts or durations of time (e.g., “1 hour, 50 minutes”)

Provide start/end dates, time interval, or date components

NSDateComponentsFormatter

For display of amounts or durations of time (e.g., “1 hour, 50 minutes”)

Provide start/end dates, time interval, or date components

Specify date and time styles and which components should be included

NSDateComponentsFormatter

For display of amounts or durations of time (e.g., “1 hour, 50 minutes”)

Provide start/end dates, time interval, or date components

Specify date and time styles and which components should be included

Locale-sensitive options to add “about” or “remaining”

NSDateComponentsFormatter

US English	French	Chinese
One hour, fifty minutes	Une heure et cinquante minutes	一小时五十分钟
1 hour, 50 minutes	1 heure et 50 minutes	1小时50分钟
1:50	1:50	1:50
About 2 hours	Environ 2 heures	约2小时

NSDateIntervalFormatter

NSDateIntervalFormatter

For display of time intervals (e.g., “9:00 AM - 4:30 PM”)

NSDateIntervalFormatter

For display of time intervals (e.g., “9:00 AM - 4:30 PM”)

Provide start and end dates

NSDateIntervalFormatter

For display of time intervals (e.g., “9:00 AM - 4:30 PM”)

Provide start and end dates

Specify date and time styles or a template

NSDateIntervalFormatter

US English	French	Chinese
June 2 - 6, 2014	2-6 juin 2014	2014年6月2日至6日
9:00 AM - 4:30 PM	9:00 - 16:30	上午9:00 - 下午4:30
June 2, 9:00 AM - June 6, 4:30 PM	2 juin 9:00 - 6 juin 16:30	6月2日 上午9:00 - 6月6日 下午4:30
June 3, 9:00 - 10:00 AM	3 juin 9:00 - 10:00	6月3日 上午9:00至10:00

Energy, Length, Mass

NEW

Energy, Length, Mass

NSEnergyFormatter, NSLengthFormatter, NSMassFormatter

Energy, Length, Mass

NSEnergyFormatter, NSLengthFormatter, NSMassFormatter

For display of energy, length, or mass with localized units

Energy, Length, Mass

NSEnergyFormatter, NSLengthFormatter, NSMassFormatter

For display of energy, length, or mass with localized units

Choose a specific unit or request a locale-appropriate unit

Energy, Length, Mass

NSEnergyFormatter, NSLengthFormatter, NSMassFormatter

For display of energy, length, or mass with localized units

Choose a specific unit or request a locale-appropriate unit

Has a number formatter for the numeric portion

Energy, Length, Mass

NSEnergyFormatter, NSLengthFormatter, NSMassFormatter

For display of energy, length, or mass with localized units

Choose a specific unit or request a locale-appropriate unit

Has a number formatter for the numeric portion

Can specify if mass is for person's weight or if energy is for food energy

Energy, Length, Mass

US English	French	Chinese
234 Cal	980 kJ	980 千焦
26.2 miles	42,2 kilomètres	42.2公里
128 lb	58 kg	58千克

Formatting Context

NEW

Formatting Context

Available for NSDateFormatter, NSNumberFormatter, NSDateComponentsFormatter, and NSByteCountFormatter

Formatting Context

Available for NSDateFormatter, NSNumberFormatter, NSDateComponentsFormatter, and NSByteCountFormatter

Formatting context property

Formatting Context

Available for NSDateFormatter, NSNumberFormatter, NSDateComponentsFormatter, and NSByteCountFormatter

Formatting context property

- NSFormattingContextStandalone

About 2 hours

Formatting Context

Available for NSDateFormatter, NSNumberFormatter, NSDateComponentsFormatter, and NSByteCountFormatter

Formatting context property

- NSFormattingContextStandalone
- NSFormattingContextListItem

About 2 hours

About 2 hours

Formatting Context

Available for NSDateFormatter, NSNumberFormatter, NSDateComponentsFormatter, and NSByteCountFormatter

Formatting context property

- NSFormattingContextStandalone — About 2 hours
- NSFormattingContextListItem — About 2 hours
- NSFormattingContextBeginningOfSentence — About 2 hours

Formatting Context

Available for NSDateFormatter, NSNumberFormatter, NSDateComponentsFormatter, and NSByteCountFormatter

Formatting context property

- NSFormattingContextStandalone — About 2 hours
- NSFormattingContextListItem — About 2 hours
- NSFormattingContextBeginningOfSentence — About 2 hours
- NSFormattingContextMiddleOfSentence — about 2 hours

Formatting Context

Available for NSDateFormatter, NSNumberFormatter, NSDateComponentsFormatter, and NSByteCountFormatter

Formatting context property

- NSFormattingContextStandalone — About 2 hours
- NSFormattingContextListItem — About 2 hours
- NSFormattingContextBeginningOfSentence — About 2 hours
- NSFormattingContextMiddleOfSentence — about 2 hours
- NSFormattingContextDynamic

Languages and Locales

Deborah Goldsmith
Software Engineer

Languages and Locales

Language

[en, zh-Hans, zh-Hant]
English, Simplified Chinese, Traditional Chinese

Locale

en_US
English (U.S.)

Two Preferences, Different Uses

User’s preferred languages

User’s regional preference

Neither is the language of user’s content

Preferred Language List

Preferred Language List

Language preference determines:

- NSBundle: which localization (.lproj) gets used
- NSString: localized comparison, text break
- WebKit: languages requested in HTTP header

Preferred Language List

Language preference determines:

- NSBundle: which localization (.lproj) gets used
- NSString: localized comparison, text break
- WebKit: languages requested in HTTP header

List is in preference order

- Apps, web sites use the first language in the list that they support
- Comparison, word break use the first language in the list

Preferred Language List

Language preference determines:

- NSBundle: which localization (.lproj) gets used
- NSString: localized comparison, text break
- WebKit: languages requested in HTTP header

List is in preference order

- Apps, web sites use the first language in the list that they support
- Comparison, word break use the first language in the list

Changing the language preference requires restarting apps

Locale (Region) Preference

Locale (Region) Preference

Locale preference determines:

- NSLocale properties
- Formatter behavior (date, number, currency, etc.)
- NSCalendar behavior

Locale (Region) Preference

Locale preference determines:

- NSLocale properties
- Formatter behavior (date, number, currency, etc.)
- NSCalendar behavior

Can change the locale preference *without* restarting apps

What’s in a Locale?

Language + optional script + region + optional keywords

```
[NSLocale currentLocale].localeIdentifier
```

What’s in a Locale?

Language + optional script + region + optional keywords

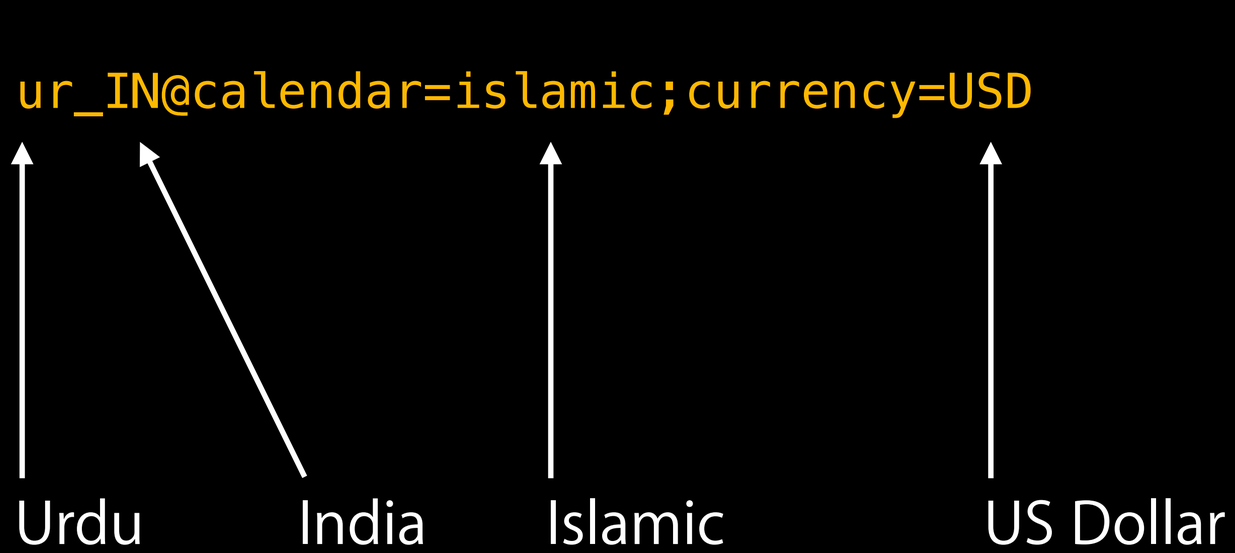
```
[NSLocale currentLocale].localeIdentifier
```

ur_IN@calendar=islamic;currency=USD

What's in a Locale?

Language + optional script + region + optional keywords

```
[NSLocale currentLocale].localeIdentifier
```



What *Shouldn't* Come from the Locale

What *Shouldn't* Come from the Locale

Physical location — use CoreLocation

What *Shouldn't* Come from the Locale

Physical location — use CoreLocation

Localization — use NSBundle: localization ≠ locale

What *Shouldn't* Come from the Locale

Physical location — use CoreLocation

Localization — use NSBundle: localization ≠ locale

Language(s) of document content — use NSLinguisticTagger to identify

What *Shouldn't* Come from the Locale

Physical location — use CoreLocation

Localization — use NSBundle: localization ≠ locale

Language(s) of document content — use NSLinguisticTagger to identify

Country of residence — credit cards, banks, etc.; must use other means

What *Shouldn't* Come from the Locale

Physical location — use CoreLocation

Localization — use NSBundle: localization ≠ locale

Language(s) of document content — use NSLinguisticTagger to identify

Country of residence — credit cards, banks, etc.; must use other means

Currency for commerce — must use other means

- [NSNumberFormatter setCurrencyCode:] lets you set it (but won't convert)

How to Read the Language/Locale Prefs

How to Read the Language/Locale Prefs

You don’t have to! This is mostly handled for you

- Formatters, NSCalendar use the correct locale by default
- NSBundle handles localization/languages
- NSString uses most-preferred language for comparison, word break

No, I *Really* Need to Know

No, I *Really* Need to Know

The language you're running in

- Use NSBundle for a localized resource from your .lproj instead
- If you get localized information from elsewhere, then needed

No, I *Really* Need to Know

The language you're running in

- Use NSBundle for a localized resource from your .lproj instead
- If you get localized information from elsewhere, then needed

The user's preferred languages

- NSBundle can handle this for you too
- +[NSBundle preferredLocalizationsFromArray:] applies .lproj logic to anything

No, I *Really* Need to Know

The language you're running in

- Use NSBundle for a localized resource from your .lproj instead
- If you get localized information from elsewhere, then needed

The user's preferred languages

- NSBundle can handle this for you too
- +[NSBundle preferredLocalizationsFromArray:] applies .lproj logic to anything

Document linguistic content

- Use NSLinguisticTagger
- User content can be multilingual!

What Language Am I Running in?

```
NSArray *preferredLocalizations =
    [[NSBundle mainBundle] preferredLocalizations];
NSString *currentLocalization = preferredLocalizations.firstObject;
```

Localized Currency Name

```
NSLocale *localizationLocale =
    [NSLocale localeWithLocaleIdentifier:currentLocalization];
NSString *yuanName = [localizationLocale
    displayNameForKey:NSLocaleCurrencyCode value:@"CNY"];
```

Localized Currency Name

```
NSLocale *localizationLocale =
    [NSLocale localeWithLocaleIdentifier:currentLocalization];
NSString *yuanName = [localizationLocale
    displayNameForKey:NSLocaleCurrencyCode value:@"CNY"];
```

English	French	Chinese
Chinese Yuan	yuan renminbi chinois	人民币

Localized Quotes

```
NSLocale *localizationLocale =
    [NSLocale localeWithLocaleIdentifier:currentLocalization];
NSString *beginQuote = [localizationLocale
    objectForKey:NSLocaleQuotationBeginDelimiterKey];
NSString *endQuote = [localizationLocale
    objectForKey:NSLocaleQuotationEndDelimiterKey];
```

Localized Quotes

```
NSLocale *localizationLocale =
    [NSLocale localeWithLocaleIdentifier:currentLocalization];
NSString *beginQuote = [localizationLocale
    objectForKey:NSLocaleQuotationBeginDelimiterKey];
NSString *endQuote = [localizationLocale
    objectForKey:NSLocaleQuotationEndDelimiterKey];
```

English	French	Japanese
“ ”	« »	「 」

Locale Change

What many apps do when user changes locale:

Locale Change

What many apps do when user changes locale:

This space left blank — unintentionally

Locale Change

What you should do:

- Listen for NSCurrentLocaleDidChangeNotification
- If you keep formatters/locales around, use +[NSLocale autoupdatingCurrentLocale]
- Redo any template formats
- Invalidate any active views that contain locale-derived information

Locale Change

```
[[NSNotificationCenter defaultCenter] addObserver:self
    selector:@selector(userChangedLocale:)
    name:NSCurrentLocaleDidChangeNotification object:nil];

…

- (void)userChangedLocale:(NSNotification *)notification {
    myFormatter.locale = [NSLocale currentLocale];
    myFormatter.dateFormat =
        [NSDateFormatter dateFormatFromTemplate:myTemplate
                                        options:0 locale:myFormatter.locale];
    [myView setNeedsDisplay];
}
```

Case Studies

Karan Misra
Software Engineer

Dates and Times

5:26 PM

Dates and Times

ཕྱི་ཉི་ 5:26

অপরাহ্ণ ৫:২৬

অ ৫:২৬

17:26

5:26 PM

م٥:٢٦

午後5:26

下午5:26

٥:٢٦ب.د.

Dates and Times

5:26 PM　　下午5:26

Dates and Times

5:26 下午

下午5:26

Dates and Times

June 3rd

Dates and Times

六月 3rd

Dates and Times

六月 3rd

6月3日

Dates and Times

2014年6月3日

Dates and Times

2014June3

Dates and Times

2014June3

June 3, 2014

Dates and Times

Don’t use explicit format strings

```
NSDateFormatter *dateFormatter = ...
[dateFormatter setDateFormat:@"MMM d, y h:mm a"];
[dateFormatter stringFromDate:[NSDate date]];
```

Locale	Date	Time
English (U.S.)	Jun 3, 2014	10:14 AM
English (U.K.)	Jun 3, 2014	10:14 am
French (France)	juin 3, 2014	10:14 AM
Chinese (China)	6月 3, 2014	10:14 上午

Dates and Times

Preset styles

```
[NSDateFormatter localizedStringFromDate:[NSDate date]
                               dateStyle:NSDateFormatterMediumStyle
                               timeStyle:NSDateFormatterShortStyle]
```

Locale	Date	Time
English (U.S.)	Jun 3, 2014	10:14 AM
English (U.K.)	3 Jun 2014	10:14
French (France)	3 juin 2014	10:14
Chinese (China)	2014年6月3日	上午10:14

Dates and Times

Custom styles

When the default formats don’t meet your needs

```
NSDateFormatter *dateFormatter = [NSDateFormatter new];
formatString = [NSDateFormatter
    dateFormatFromTemplate:@"dMMM"
                   options:0
                    locale:dateFormatter.locale];
[dateFormatter setDateFormat:formatString];
```

Dates and Times

Really custom styles

When your designer insists that it must look a certain way

```
NSLocale *locale = [NSLocale currentLocale];
NSString *language = [locale objectForKey:NSLocaleLanguageCode];
NSString *region = [locale objectForKey:NSLocaleCountryCode];
if ([language isEqualToString:@"en"] && [region isEqualToString:@"US"]) {
    // use custom format for English (US)
}
else if ([language isEqualToString:@"ja"]) {
    // use custom format for Japanese
}
else {
    // use preset style or template
}
```

Names

Johnny Appleseed

Names

Benoît Sokal

करन मिश्र

林黛玉

Johnny Appleseed

فاروق شیخ

김태연

スティーブ・ジョブス

Names

AppleseedJohnny

Names

黛玉 林

Names

Don’t assume “First Last”

```
NSString *displayName = [NSString stringWithFormat:
                            @"%@ %@", firstName, lastName];
```

Names

Don’t use a fixed format

```
NSString *formatString = /* localized string */
NSString *displayName = [NSString stringWithFormat:
                            formatString, firstName, lastName];
```

Names

ABRecordCopyCompositeName

```
ABRecordRef record = /* get record from AB */
NSString *displayName = (NSString *)ABRecordCopyCompositeName(record);
```

Names

ABRecordCopyCompositeName

```
NSString *firstName = ...
NSString *lastName = ...
ABRecordRef record = ABPersonCreate();
ABRecordSetValue(record, kABPersonFirstNameProperty,
                                    (CFStringRef)firstName, NULL);
ABRecordSetValue(record, kABPersonLastNameProperty,
                                    (CFStringRef)lastName, NULL);
NSString *displayName = (NSString *)ABRecordCopyCompositeName(record);
CFRelease(record);
```

Names

-[ABPerson displayName]

```
@interface ABPerson : ABRecord
@property (readonly) NSString *displayName;
@end
```

Right-to-Left Text

مرحباً שלום

السلامُ عليكم

Right-to-Left Text

مرحباً שלום

السلامُ عليكم

Text Alignment

```
[textView setAlignment:NSTextAlignmentLeft]
```

Text Alignment

Einstein's “Zur Elektrodynamik bewegter Körper” (“On the Electrodynamics of Moving Bodies”) was received on 30 June 1905. It reconciles Maxwell’s equations for electricity and magnetism with the laws of mechanics, by introducing major changes to mechanics close to the speed of light. This later became known as Einstein's special theory of relativity.

ورقة أينشتاين العلمية الثالثة كانت عن "النظرية النسبية الخاصة" ، فتناولت الورقة الزمان ، والمكان ، والكتلة ، والطاقة ، وأسهمت نظرية أينشتاين بإزالة الغموض الذي نجم عن التجربة الشهيرة التي أجراها الأمريكيان الفيزيائي ألبرت ميكلسون والكيميائي إدوارد مورلي

أثبت أينشتاين أن موجات الضوء تستطيع أن تنتشر في الخلاء دون الحاجة لوجود وسط أو مجال ، على خلاف الموجات الأخرى المعروفة التي تحتاج إلى وسط تنتشر فيه كالهواء أو الماء وأن سرعة الضوء هي سرعة ثابتة وليست نسبية مع (الملاحظ حركة المراقب)

Text Alignment

Use natural alignment

```
[textView setAlignment:NSTextAlignmentNatural]
```

Text Alignment

Use natural alignment

Einstein's “Zur Elektrodynamik bewegter Körper” (“On the Electrodynamics of Moving Bodies”) was received on 30 June 1905.

It reconciles Maxwell’s equations for electricity
and magnetism with the laws of mechanics, by introducing major changes to mechanics close to the speed of light. This later became known as Einstein's special theory of relativity.

ورقة أينشتاين العلمية الثالثة كانت عن "النظرية النسبية الخاصة" ، فتناولت الورقة الزمان ، والمكان ، والكتلة ، والطاقة ، وأسهمت نظرية أينشتاين بإزالة الغموض الذي نجم عن التجربة الشهيرة التي أجراها الأمريكيان الفيزيائي ألبرت ميكلسون والكيميائي إدوارد مورلي

أثبت أينشتاين أن موجات الضوء تستطيع أن تنتشر في الخلاء دون الحاجة لوجود وسط أو مجال ، على خلاف الموجات الأخرى المعروفة التي تحتاج إلى وسط تنتشر فيه كالهواء أو الماء وأن سرعة الضوء هي سرعة ثابتة وليست نسبية مع (حركة المراقب (الملاحظ

Text Directionality

```
[textField setBaseWritingDirection:NSWritingDirectionRightToLeft
                             forRange:range]
```

Text Directionality

WWDC is a مست conference

“conference fun WWDC is a”

Text Directionality

Use natural writing direction

```
[textField setBaseWritingDirection:NSWritingDirectionNatural
                          forRange:range]
```

Text Directionality

Use natural writing direction

WWDC is a مست conference

"WWDC is a fun conference"

Text Directionality

When things get complicated

Text Directionality

When things get complicated

Safari موجود على Mac

“Mac on Safari”

Text Directionality

When things get complicated

Text Directionality

When things get complicated

“Safari on Mac”

Text Directionality

When things get complicated

“Safari on Mac”

Text Directionality

When things get complicated

Safari موجود على Mac

RLM
U+200F

“Safari on Mac”

Text Directionality

When things get complicated

Text Directionality

When things get complicated

شكراً means “Thank you”

Text Directionality

When things get complicated

Text Directionality

When things get complicated

LRM
U+200E

شكراً means “Thank you”

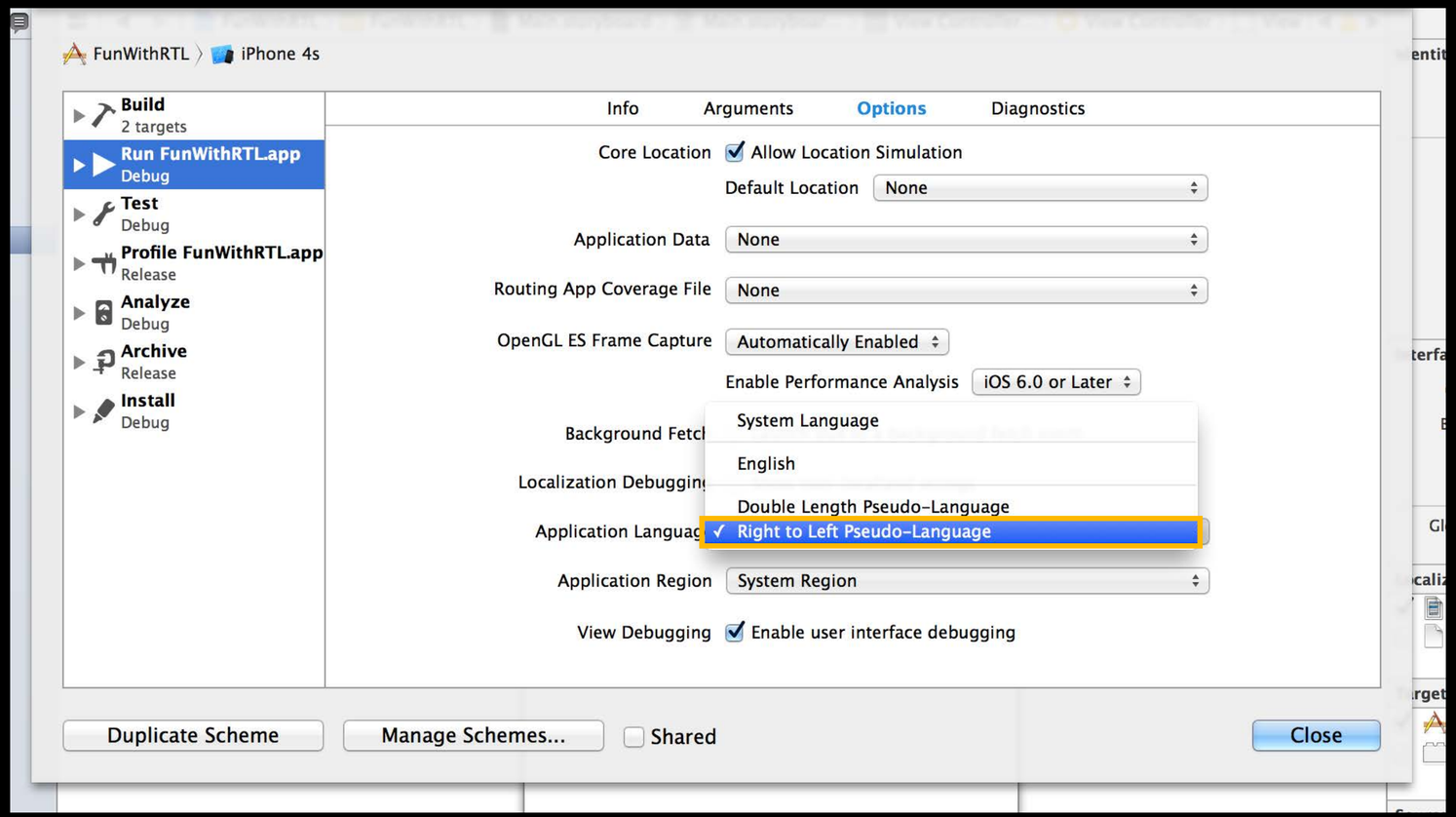
Text Directionality

When things get complicated

Refer to “Internationalization and Localization Guide” for more information

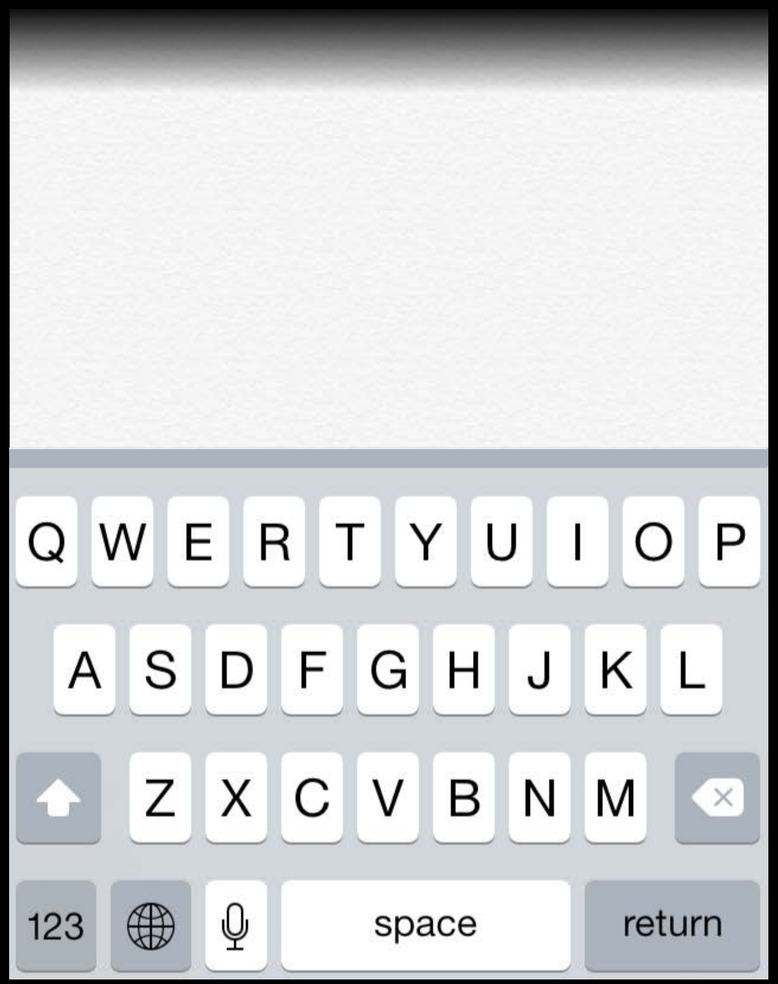
Simulating Right-to-Left

“Right-to-left Pseudo-Language” in Xcode 6



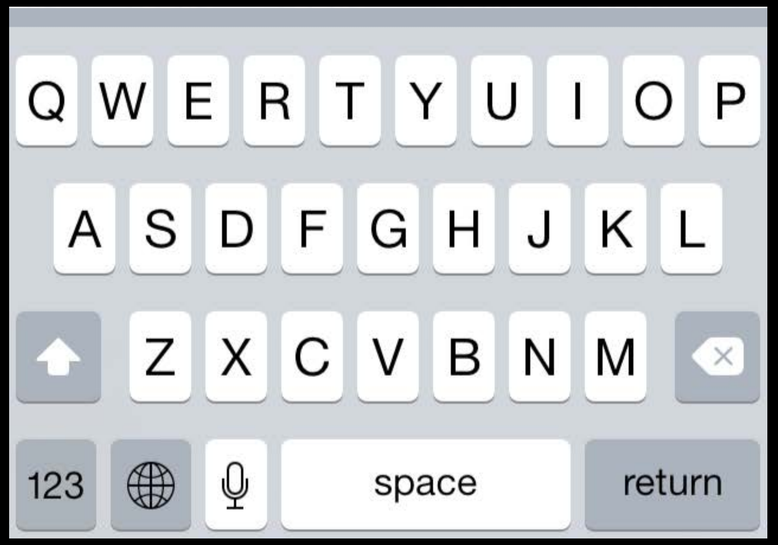
Localizing with Xcode 6 — Marina — Tuesday 11:30AM

Keyboard Size and Position



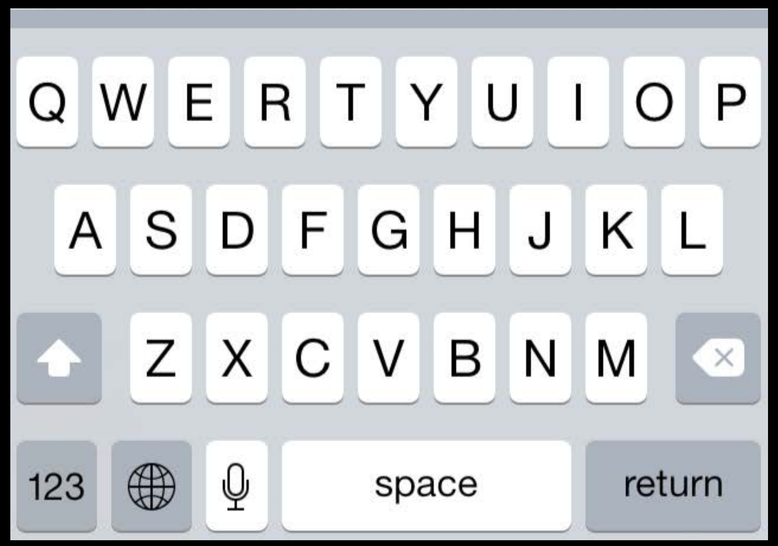
Keyboard Size and Position

English



Keyboard Size and Position

English



Predictive: Off

Predictive: On

Keyboard Size and Position

Bengali and English

Bengali

English

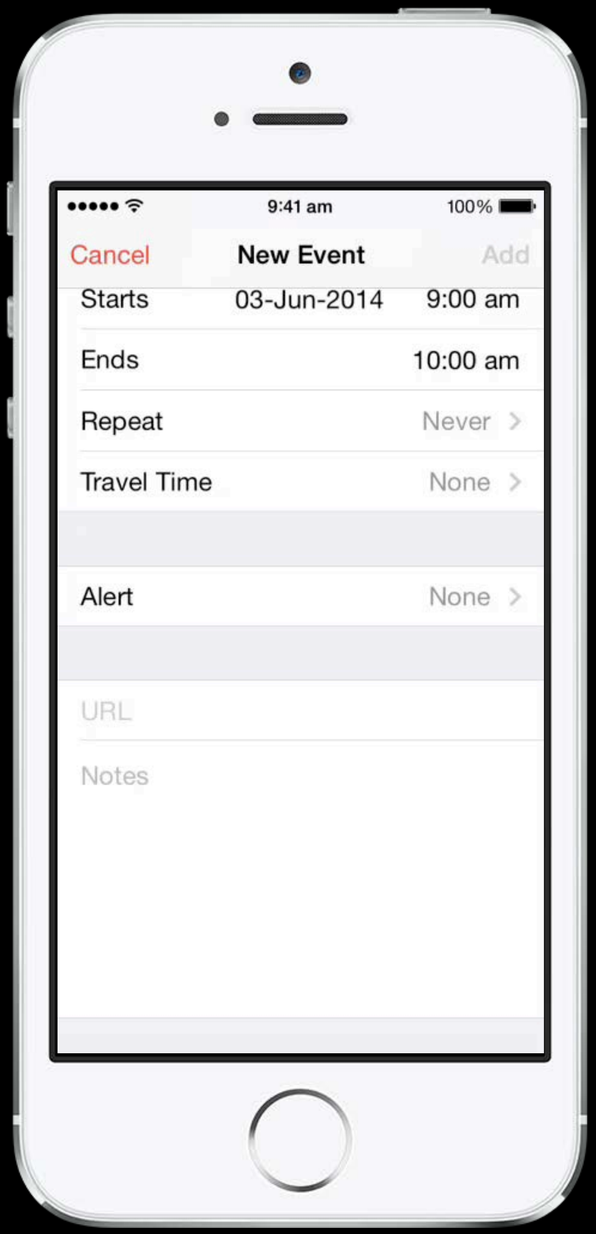
Keyboard Size and Position

Japanese Kana

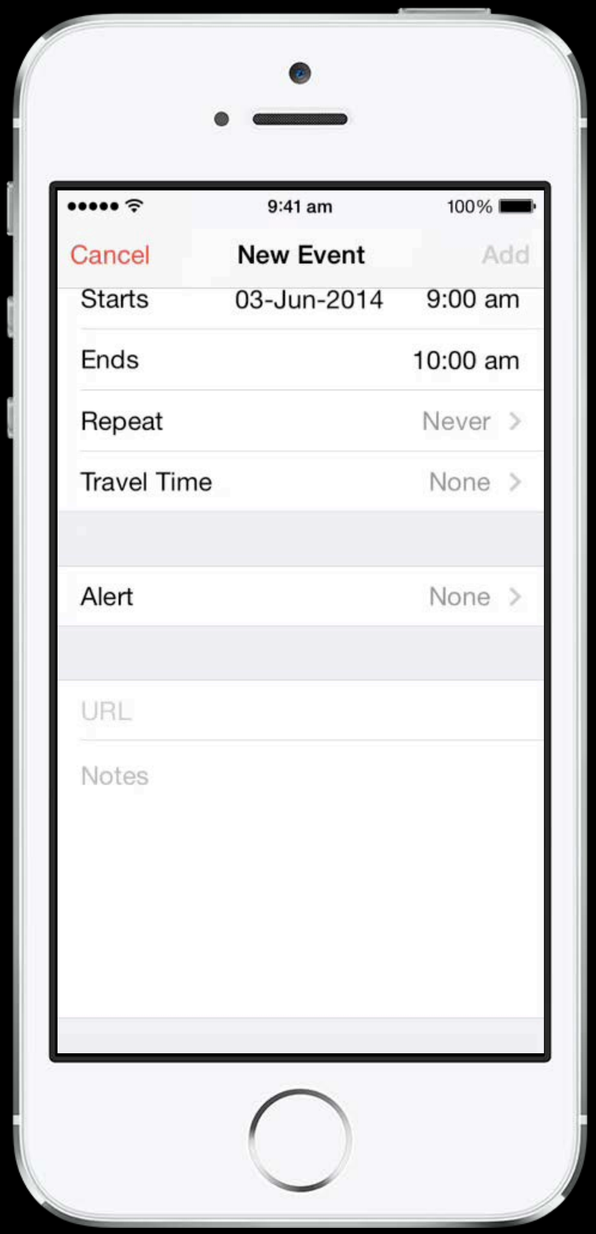
Before Typing

While Typing

Keyboard Size and Position



Keyboard Size and Position



Keyboard Size and Position

Responding to changes

Keyboard Appearing

`UIKeyboardWillShowNotification`
`UIKeyboardDidShowNotification`

Keyboard Disappearing

`UIKeyboardWillHideNotification`
`UIKeyboardDidHideNotification`

Keyboard Resizing

`UIKeyboardWillChangeFrameNotification`
`UIKeyboardDidChangeFrameNotification`

Keyboard Size and Position

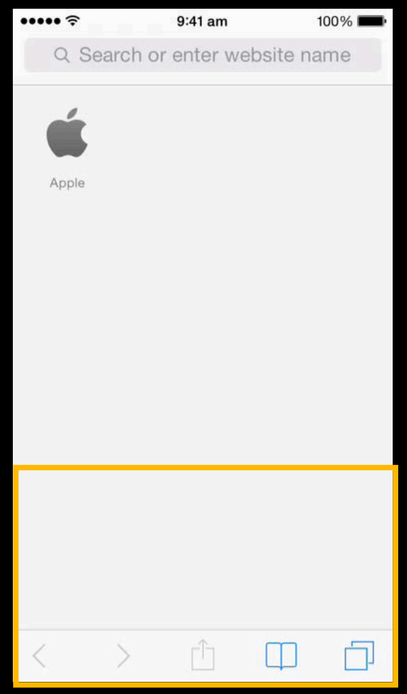
Responding to changes

Keyboard Size and Position

Responding to changes

Frame

`UIKeyboardFrameBeginUserInfoKey`
`UIKeyboardFrameEndUserInfoKey`



Keyboard Size and Position

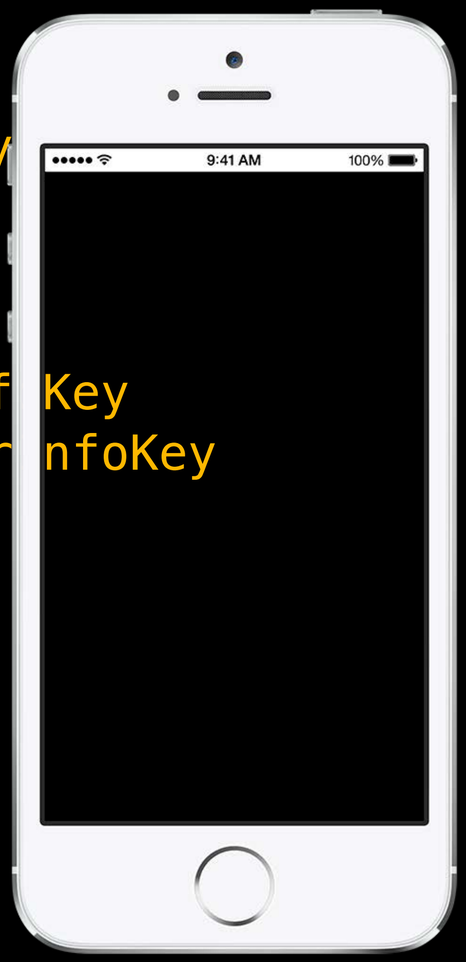
Responding to changes

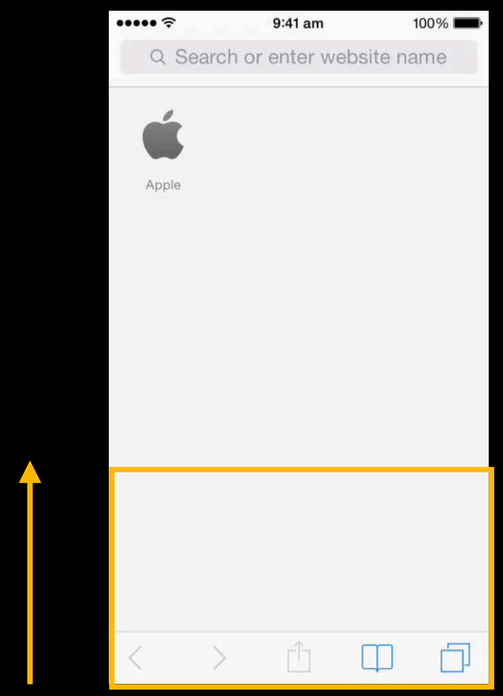
Frame

`UIKeyboardFrameBeginUserInfoKey`
`UIKeyboardFrameEndUserInfoKey`

Animation

`UIKeyboardAnimationCurveUserInf Key`
`UIKeyboardAnimationDurationUser nfoKey`

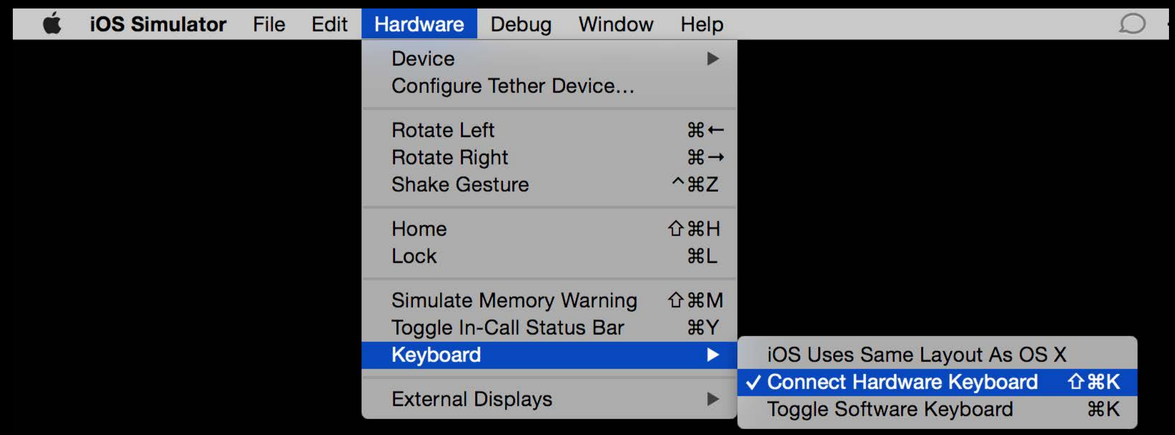




Hardware Keyboards

Bluetooth keyboards can be used with iPhone, iPad and iPod touch

iOS Simulator can simulate hardware keyboard typing



Hardware Keyboards

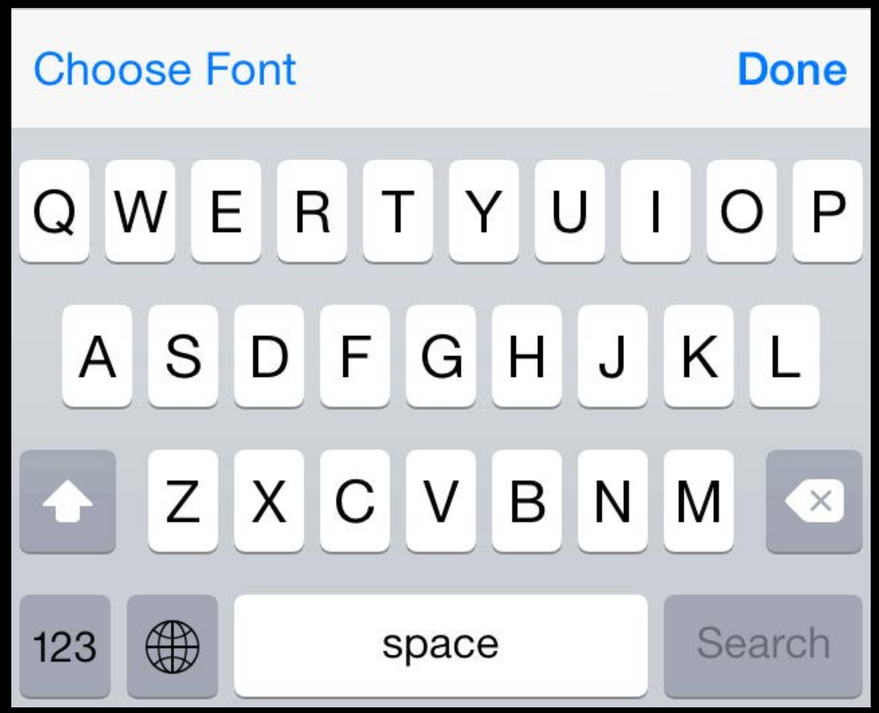
Hardware Keyboards

Hardware Keyboards

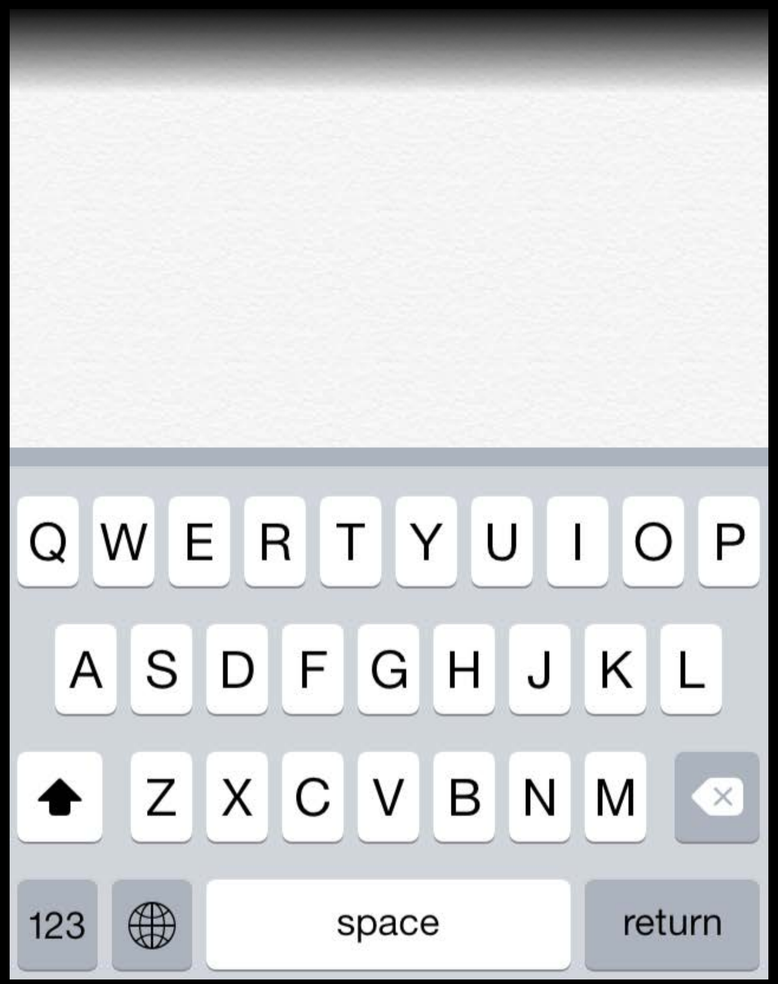
Hardware Keyboards

Input Accessory Views

```
textView.inputAccessoryView = accessoryView;
```



Keyboard Type



Keyboard Type

English

Different keyboard types optimize for different kinds of text entry, such as Email, URL, Twitter, Search, etc.

Keyboard Type

English

Different keyboard types optimize for different kinds of text entry, such as Email, URL, Twitter, Search, etc.

Default

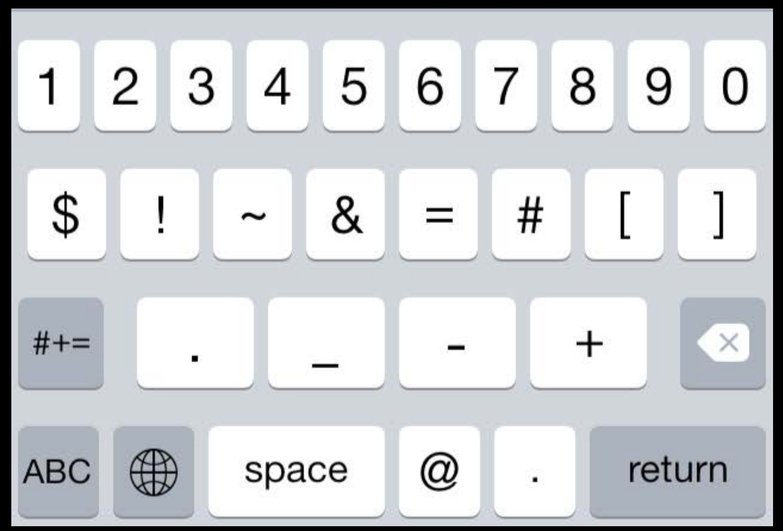
Email

Keyboard Type

English

Different keyboard types optimize for different kinds of text entry, such as Email, URL, Twitter, Search, etc.

Default



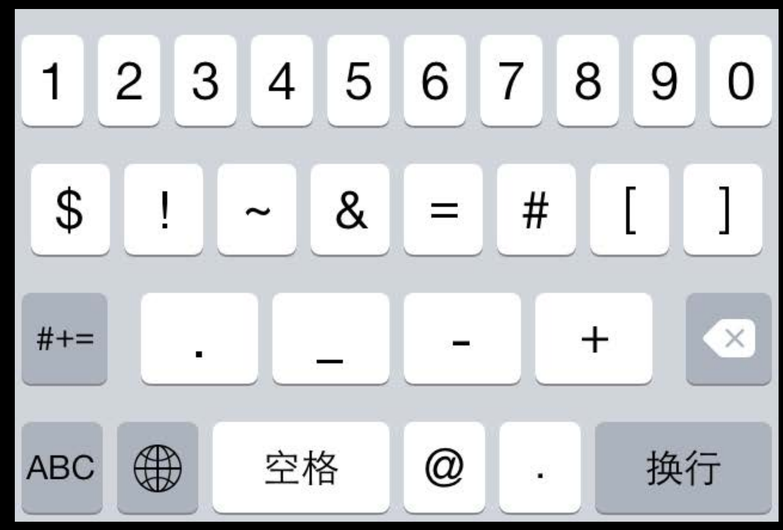
Email

Keyboard Type

Simplified Chinese

Even more important for languages in which the default set of punctuation symbols is different from that which is used for typing email addresses, URLs, etc.

Default



Email

Keyboard Type

And there are many more!

Keyboard Type

And there are many more!

URL

Twitter

Number Pad

Keyboard Type

```
textView.keyboardType =  UIKeyboardTypeDefault
                         UIKeyboardTypeASCIICapable
                         UIKeyboardTypeNumbersAndPunctuation
                         UIKeyboardTypeURL
                         UIKeyboardTypeEmailAddress
                         UIKeyboardTypeTwitter
                         UIKeyboardTypeWebSearch
                         UIKeyboardTypeNumberPad
                         UIKeyboardTypePhonePad
                         UIKeyboardTypeNamePhonePad
                         UIKeyboardTypeDecimalPad
```

Autocorrection

```
textView.autocorrectionType =  UITextAutocorrectionTypeDefault
                               UITextAutocorrectionTypeNo
                               UITextAutocorrectionTypeYes
```

Turn off autocorrection if the user needs to type usernames, email addresses, etc.
that would not benefit from and would be impeded by it

Autocapitalization

```
textView.autocapitalizationType =
    UITextAutocapitalizationTypeNone
    UITextAutocapitalizationTypeWords
    UITextAutocapitalizationTypeSentences
    UITextAutocapitalizationTypeAllCharacters
```

If the correct autocapitalization type is specified the user can often type words properly capitalized without ever using the Shift key

Autocapitalization

```
textView.autocapitalizationType =
    UITextAutocapitalizationTypeNone
    UITextAutocapitalizationTypeWords
    UITextAutocapitalizationTypeSentences
    UITextAutocapitalizationTypeAllCharacters
```

john@apple.com
John Appleseed
This is text
WWDC

If the correct autocapitalization type is specified the user can often type words properly capitalized without ever using the Shift key

Marked Text

Intermediate or incomplete form of input

Used for Japanese and Chinese keyboards

Marked Text

Intermediate or incomplete form of input

Used for Japanese and Chinese keyboards

huan ying lai dao WWDC

input

Marked Text

Intermediate or incomplete form of input

Used for Japanese and Chinese keyboards

huan ying lai dao WWDC

input

Marked Text

Intermediate or incomplete form of input

Used for Japanese and Chinese keyboards

huan ying lai dao WWDC

input

欢迎来到WWDC

output

Marked Text

Correct handling

```
@protocol UITextInput <UIKeyInput>
@property (nonatomic, readonly) UITextRange *markedTextRange;
@end
```

Check for existence of marked text using -markedTextRange

Do not modify the document while marked text is present, since it interrupts input

Marked text is not the intended input in most cases

May be used for live search

Summary

Summary

Localize your app into many new languages

Summary

Localize your app into many new languages

Understand the differences between language and locale

Summary

Localize your app into many new languages

Understand the differences between language and locale

Format dates, times, and names correctly

Summary

Localize your app into many new languages

Understand the differences between language and locale

Format dates, times, and names correctly

Support right-to-left text whether you have a localization or not

Summary

Localize your app into many new languages

Understand the differences between language and locale

Format dates, times, and names correctly

Support right-to-left text whether you have a localization or not

Keyboards come in different sizes and change size on the fly

Summary

Localize your app into many new languages

Understand the differences between language and locale

Format dates, times, and names correctly

Support right-to-left text whether you have a localization or not

Keyboards come in different sizes and change size on the fly

Choose the right keyboard type for the context

Summary

Localize your app into many new languages

Understand the differences between language and locale

Format dates, times, and names correctly

Support right-to-left text whether you have a localization or not

Keyboards come in different sizes and change size on the fly

Choose the right keyboard type for the context

Don’t modify the document if it contains marked text

More Information

Jake Behrens
App Frameworks Evangelist
behrens@apple.com

Internationalization and Localization Guide
http://developer.apple.com

Apple Developer Forums
http://devforums.apple.com

Related Sessions

Localizing with Xcode 6	Marina	Tuesday 11:30AM
Making Your App World-Ready		WWDC 2013

Labs

Xcode and Localization Lab	Tools Lab C	Tuesday 2:00PM
Internationalization Lab	Frameworks Lab B	Tuesday 3:15PM

WWDC14