Using Fonts with Text Kit

Session 223

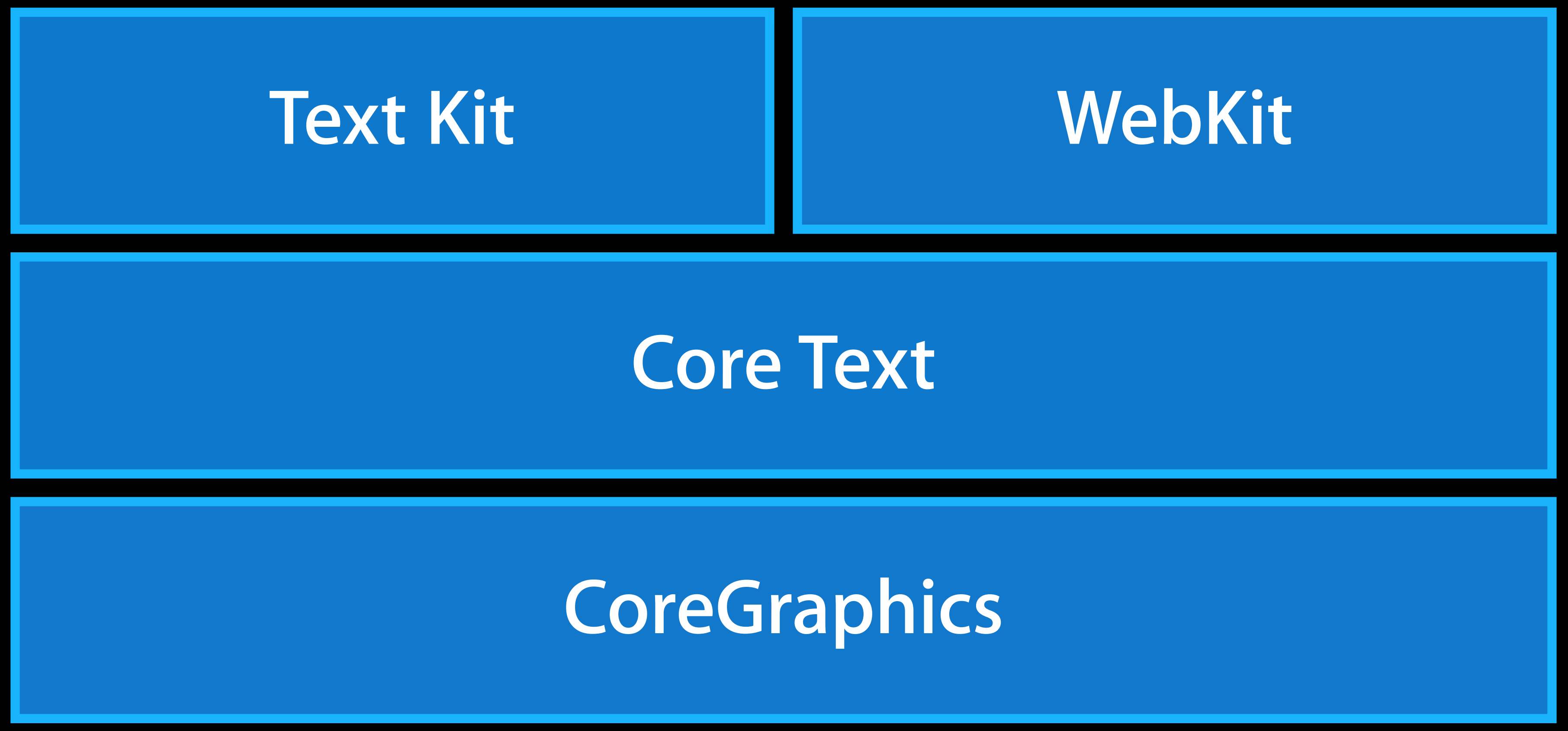
Ned Holbrook
Typographic Engineer

What You’ll Learn

- Specifying fonts
    - Dynamic Type
    - Font descriptors
    - Adding fonts
- Language tagging
- Modifying fallbacks

What Is Text Kit?

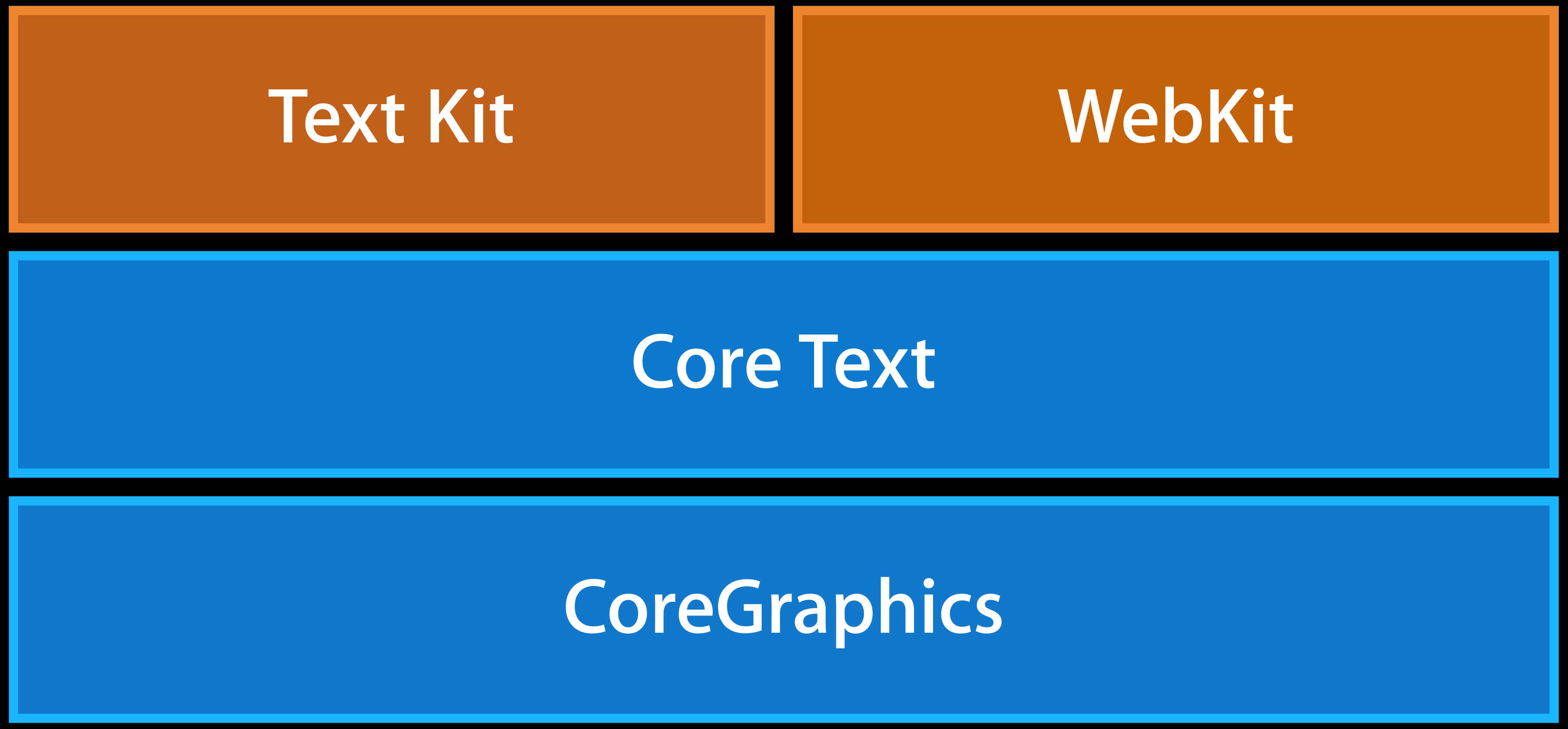
Synopsis



What Is Text Kit?

Synopsis

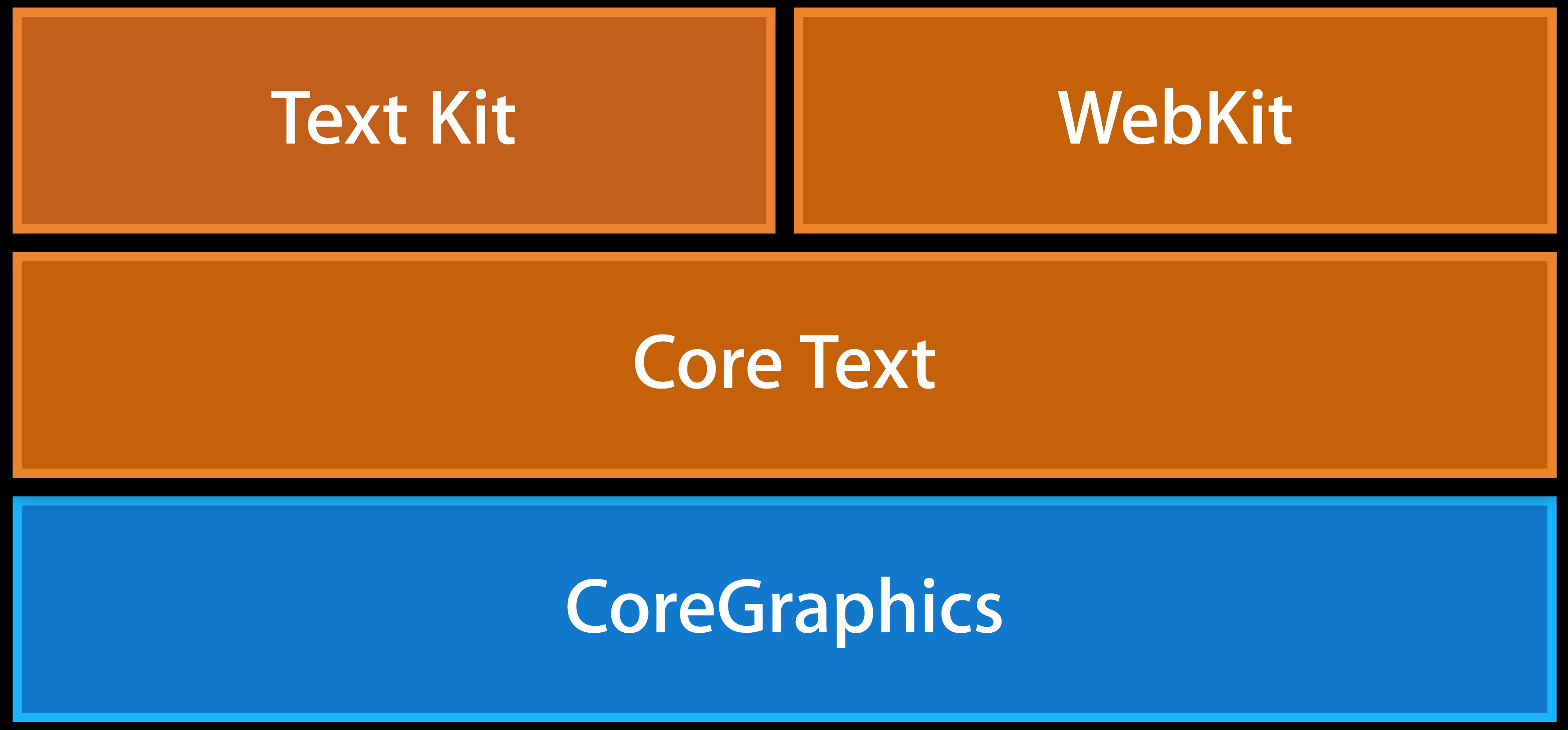
- Text Kit and Web Kit are built on Core Text



What Is Text Kit?

Synopsis

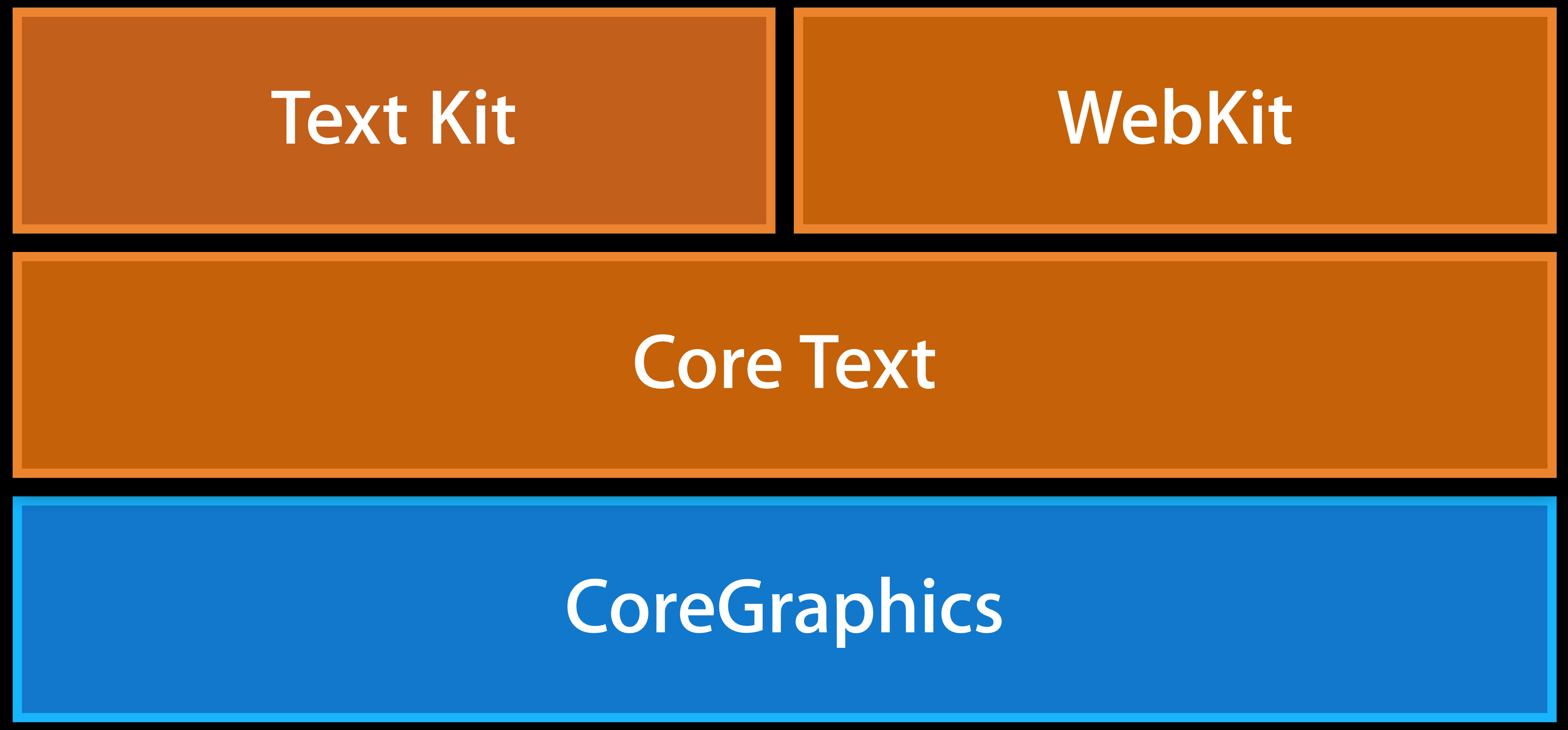
- Text Kit and Web Kit are built on Core Text
- Core Text is available for specialized tasks



What Is Text Kit?

Synopsis

- Text Kit and Web Kit are built on Core Text
- Core Text is available for specialized tasks
- Unicode text layout
  - String + Font = Glyphs

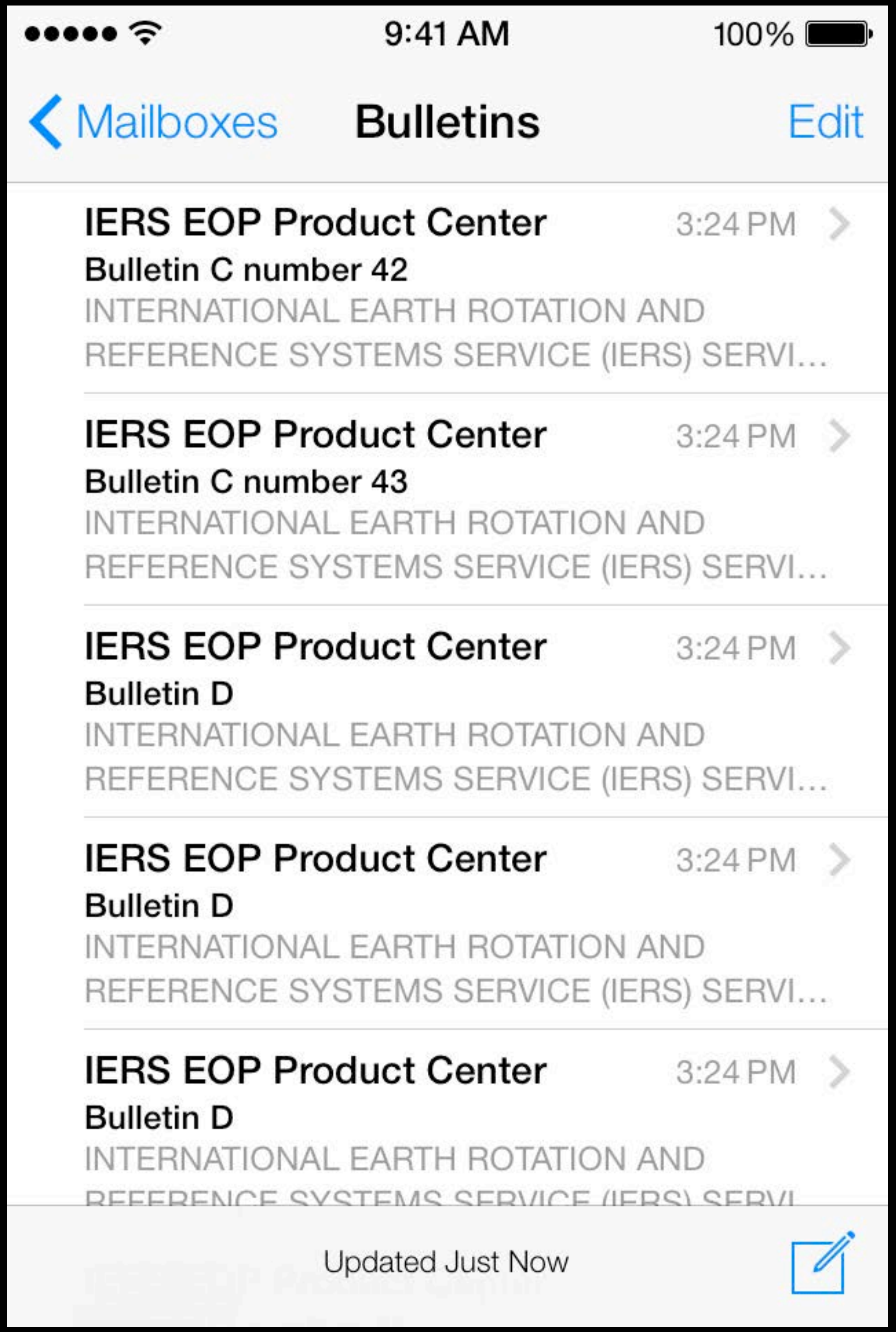


Specifying Fonts

Which Font?

- Dynamic Type
- System font
- UIFontDescriptor
- Adding fonts

Dynamic Type



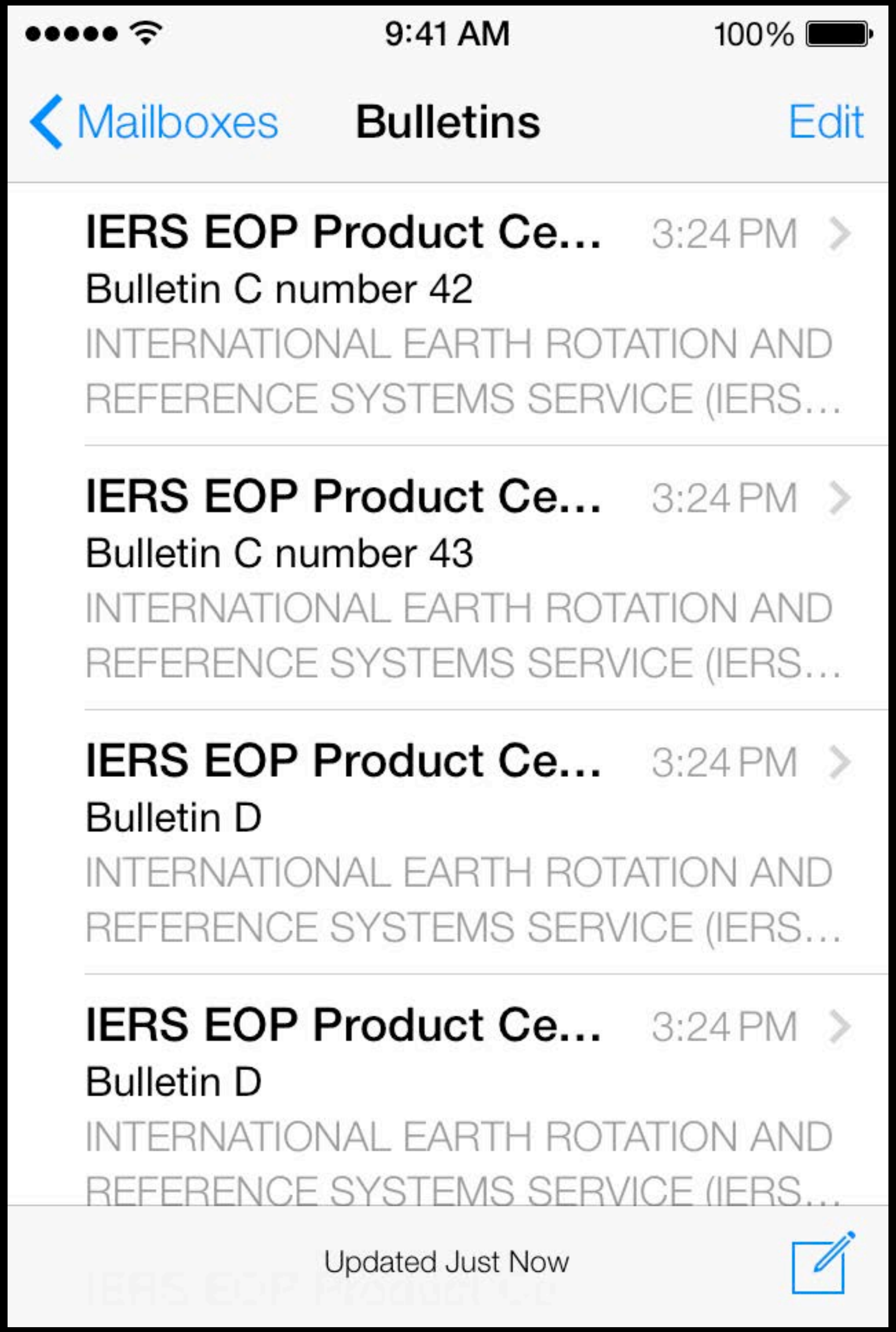
IERS EOP Product Center 3:24 PM
Bulletin C number 42
INTERNATIONAL EARTH ROTATION AND REFERENCE SYSTEMS SERVICE (IERS) SERVI...

IERS EOP Product Center 3:24 PM
Bulletin C number 43
INTERNATIONAL EARTH ROTATION AND REFERENCE SYSTEMS SERVICE (IERS) SERVI...

IERS EOP Product Center 3:24 PM
Bulletin D
INTERNATIONAL EARTH ROTATION AND REFERENCE SYSTEMS SERVICE (IERS) SERVI...

IERS EOP Product Center 3:24 PM
Bulletin D
INTERNATIONAL EARTH ROTATION AND REFERENCE SYSTEMS SERVICE (IERS) SERVI...

IERS EOP Product Center 3:24 PM
Bulletin D
INTERNATIONAL EARTH ROTATION AND REFERENCE SYSTEMS SERVICE (IERS) SERVI

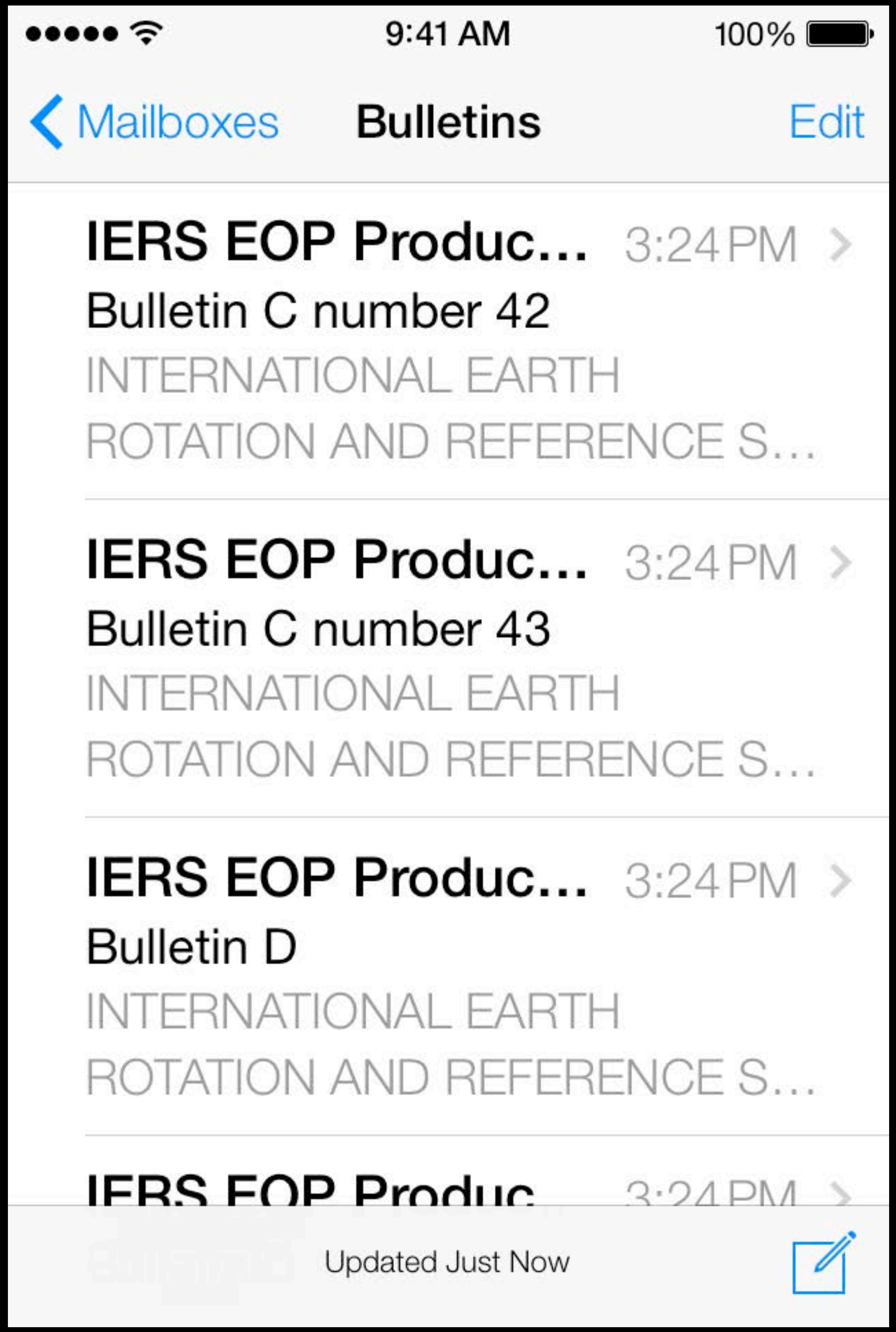


IERS EOP Product Ce... 3:24 PM
Bulletin C number 42
INTERNATIONAL EARTH ROTATION AND REFERENCE SYSTEMS SERVICE (IERS...

IERS EOP Product Ce... 3:24 PM
Bulletin C number 43
INTERNATIONAL EARTH ROTATION AND REFERENCE SYSTEMS SERVICE (IERS...

IERS EOP Product Ce... 3:24 PM
Bulletin D
INTERNATIONAL EARTH ROTATION AND REFERENCE SYSTEMS SERVICE (IERS...

IERS EOP Product Ce... 3:24 PM
Bulletin D
INTERNATIONAL EARTH ROTATION AND REFERENCE SYSTEMS SERVICE (IERS...



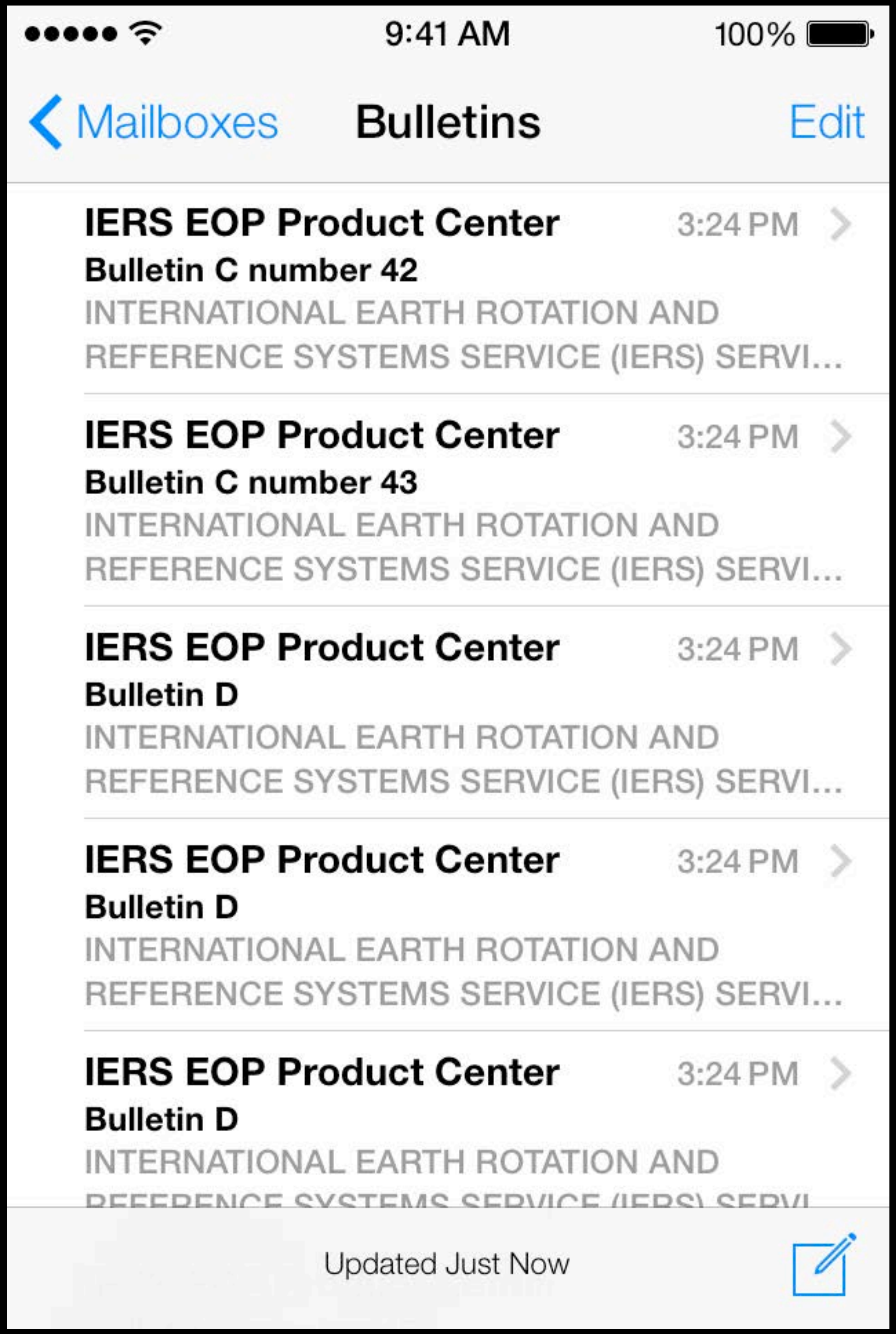
IERS EOP Produc... 3:24 PM
Bulletin C number 42
INTERNATIONAL EARTH ROTATION AND REFERENCE S...

IERS EOP Produc... 3:24 PM
Bulletin C number 43
INTERNATIONAL EARTH ROTATION AND REFERENCE S...

IERS EOP Produc... 3:24 PM
Bulletin D
INTERNATIONAL EARTH ROTATION AND REFERENCE S...

IERS EOP Produc 3:24 PM

Dynamic Type



IERS EOP Product Center 3:24 PM
Bulletin C number 42
INTERNATIONAL EARTH ROTATION AND REFERENCE SYSTEMS SERVICE (IERS) SERVI…

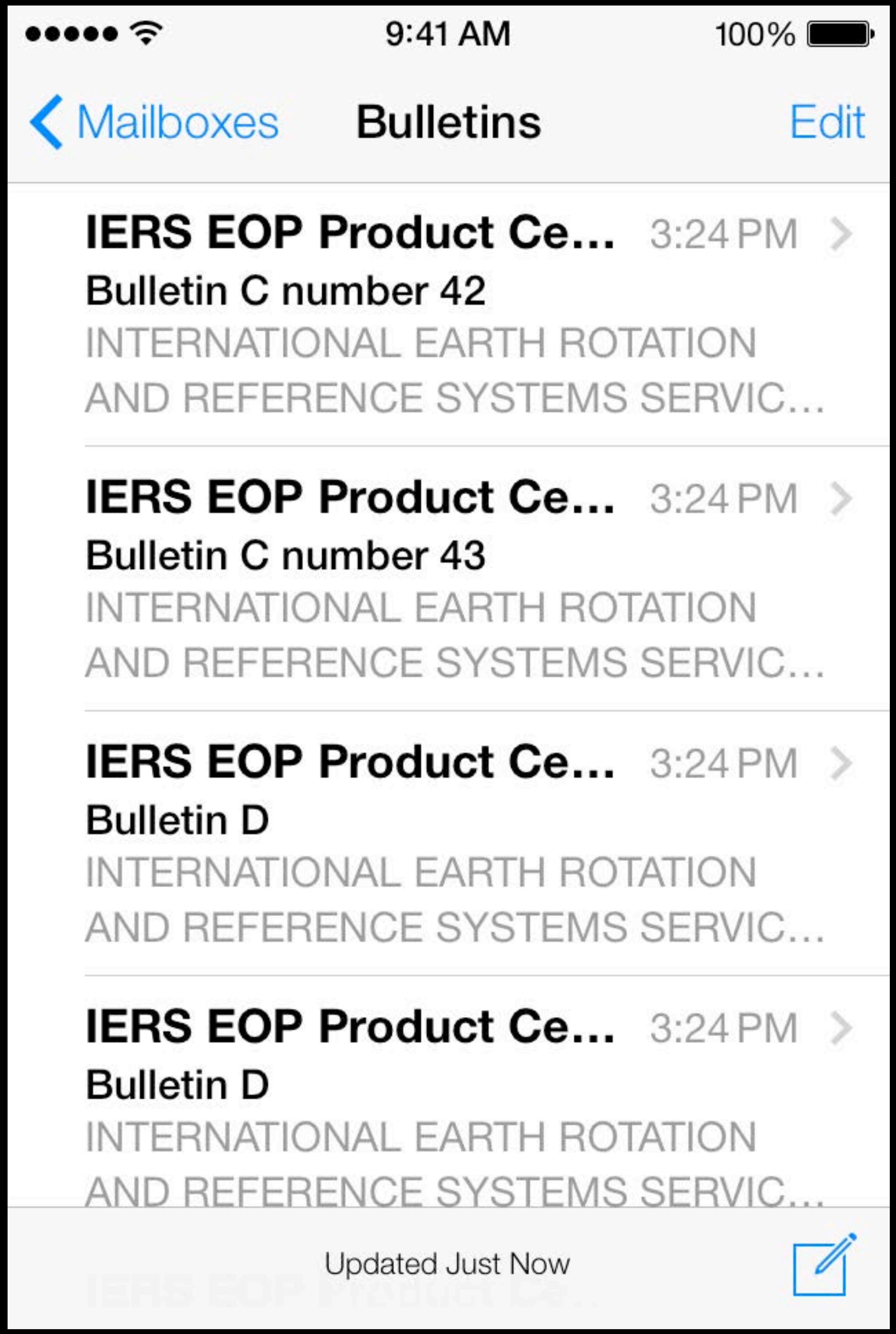
IERS EOP Product Center 3:24 PM
Bulletin C number 43
INTERNATIONAL EARTH ROTATION AND REFERENCE SYSTEMS SERVICE (IERS) SERVI…

IERS EOP Product Center 3:24 PM
Bulletin D
INTERNATIONAL EARTH ROTATION AND REFERENCE SYSTEMS SERVICE (IERS) SERVI…

IERS EOP Product Center 3:24 PM
Bulletin D
INTERNATIONAL EARTH ROTATION AND REFERENCE SYSTEMS SERVICE (IERS) SERVI…

IERS EOP Product Center 3:24 PM
Bulletin D
INTERNATIONAL EARTH ROTATION AND REFERENCE SYSTEMS SERVICE (IERS) SERVI…

Updated Just Now



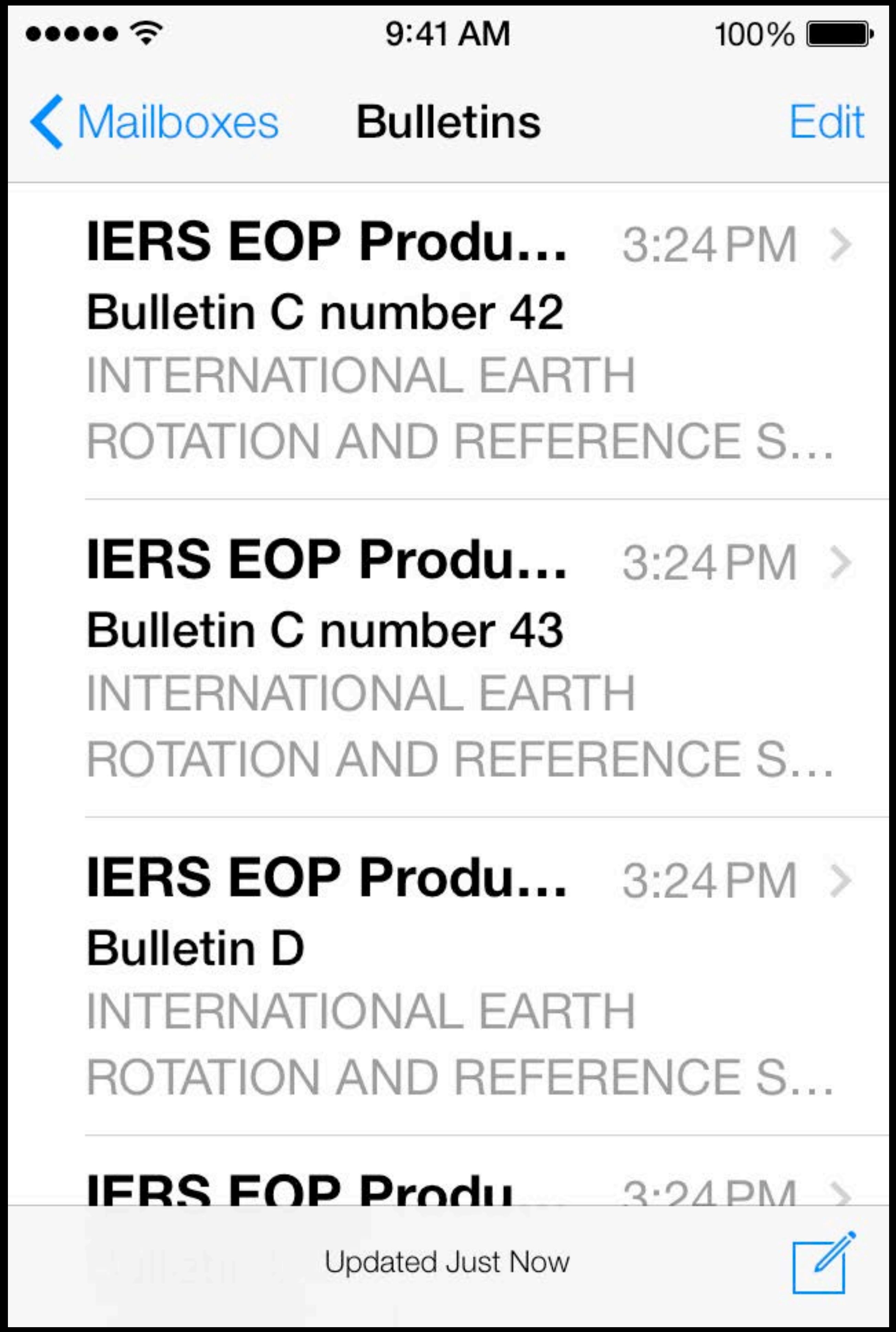
IERS EOP Product Ce… 3:24 PM
Bulletin C number 42
INTERNATIONAL EARTH ROTATION AND REFERENCE SYSTEMS SERVIC…

IERS EOP Product Ce… 3:24 PM
Bulletin C number 43
INTERNATIONAL EARTH ROTATION AND REFERENCE SYSTEMS SERVIC…

IERS EOP Product Ce… 3:24 PM
Bulletin D
INTERNATIONAL EARTH ROTATION AND REFERENCE SYSTEMS SERVIC…

IERS EOP Product Ce… 3:24 PM
Bulletin D
INTERNATIONAL EARTH ROTATION AND REFERENCE SYSTEMS SERVIC…

Updated Just Now



IERS EOP Produ… 3:24 PM
Bulletin C number 42
INTERNATIONAL EARTH ROTATION AND REFERENCE S…

IERS EOP Produ… 3:24 PM
Bulletin C number 43
INTERNATIONAL EARTH ROTATION AND REFERENCE S…

IERS EOP Produ… 3:24 PM
Bulletin D
INTERNATIONAL EARTH ROTATION AND REFERENCE S…

IERS EOP Produ… 3:24 PM

Updated Just Now

Dynamic Type

- Organized by usage
- Optimized for legibility
- User picks size
    - Auto Layout

Text Styles

Subhead comparison

Quo usque tandem abutere,
Catilina, patientia nostra?

Quo usque tandem abutere,
Catilina, patientia nostra?

Quo usque tandem abutere,
Catilina, patientia nostra?

Quo usque tandem abutere,
Catilina, patientia nostra?

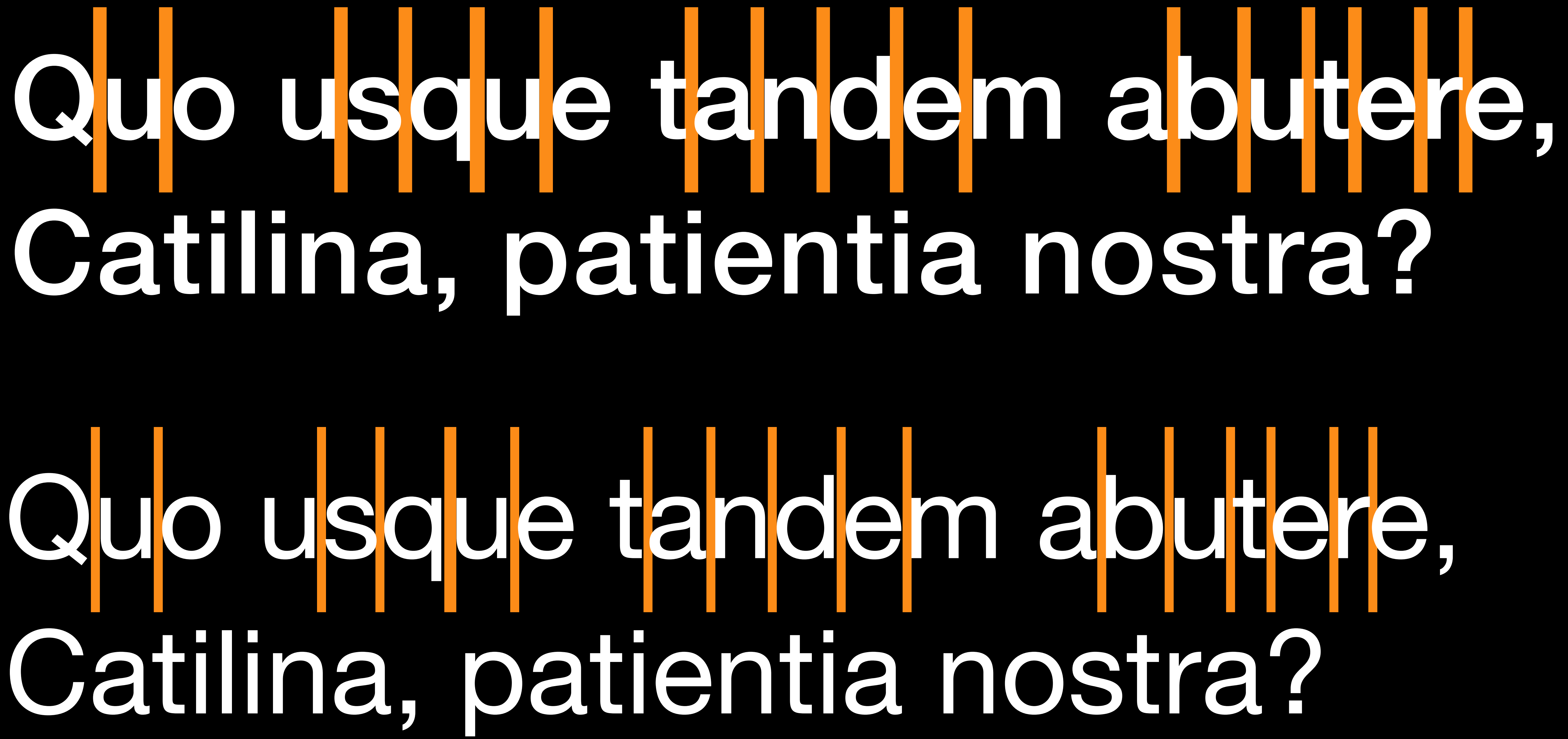
Quo usque tandem abutere,
Catilina, patientia nostra?

Quo usque tandem abutere,
Catilina, patientia nostra?

Quo usque tandem abutere,
Catilina, patientia nostra?

Text Styles

Subhead comparison



Quo usque tandem abutere,
Catilina, patientia nostra?

Quo usque tandem abutere,
Catilina, patientia nostra?

Using Text Styles

`+[UIFont preferredFontForTextStyle:]`

  - Styles in UIFontDescriptor documentation

- Can be modified with symbolic traits
- CSS properties
  - Prefixed with `-apple-system-`

System Fonts

- “Meta” fonts
  - May change across releases (e.g., Lucida Grande)
  - Hidden names start with ‘.’ (period)
- `+[UIFont systemFontOfSize:]` and friends
  - Not for user content
  - Only recompiled code will get new font on iOS 7

UIFontDescriptor

- Describes a font
- Access and modify font properties
- List available fonts
- Specify your own fonts

Descriptor Basics

- Font attributes
    - Names, size, traits, features, etc.
- Described font may not exist
- Describe an actual font by matching on attributes

Descriptor Matching

- Explicit matching

```
–[UIFontDescriptor matchingFontDescriptorsWithMandatoryKeys:]
```

- Implicit matching

```
–[UIFontDescriptor fontDescriptorWithSymbolicTraits:]
–[UIFont fontWithDescriptor:size:]
```

Descriptor Matching

Family name matching

```
UIFontDescriptor *helveticaNeueFamily =
  [UIFontDescriptor fontDescriptorWithFontAttributes: @{
    UIFontDescriptorFamilyAttribute: @"Helvetica Neue"
  }];

NSArray *matches =
  [helveticaNeueFamily matchingFontDescriptorsWithMandatoryKeys: nil];
```

Descriptor Matching

Family name matching

```
UIFontDescriptor *helveticaNeueFamily =
  [UIFontDescriptor fontDescriptorWithFontAttributes: @{
    UIFontDescriptorFamilyAttribute: @"Helvetica Neue"
  }];
```

```
NSArray *matches =
  [helveticaNeueFamily matchingFontDescriptorsWithMandatoryKeys: nil];
```

Descriptor Matching

Family name matching

```
UIFontDescriptor *helveticaNeueFamily =
  [UIFontDescriptor fontDescriptorWithFontAttributes: @{
    UIFontDescriptorFamilyAttribute: @"Helvetica Neue"
  }];
```

```
NSArray *matches =
  [helveticaNeueFamily matchingFontDescriptorsWithMandatoryKeys: nil];
```

Descriptor Matching

Family name matching

```
UIFontDescriptor *helveticaNeueFamily =
  [UIFontDescriptor fontDescriptorWithFontAttributes: @{
    UIFontDescriptorFamilyAttribute: @"Helvetica Neue"
  }];
```

```
NSArray *matches =
  [helveticaNeueFamily matchingFontDescriptorsWithMandatoryKeys: nil];
```

HelveticaNeue
HelveticaNeue-Medium
HelveticaNeue-Light
HelveticaNeue-Thin
...

Descriptor Matching

Family name matching

```
UIFontDescriptor *helveticaNeueFamily =
  [UIFontDescriptor fontDescriptorWithFontAttributes: @{
    UIFontDescriptorFamilyAttribute: @"Helvetica Neue"
  }];

NSArray *matches =
  [helveticaNeueFamily matchingFontDescriptorsWithMandatoryKeys: nil];
```

Descriptor Matching

Text style emboldening

```
UIFontDescriptor *fontDescriptor =
  [UIFontDescriptor preferredFontDescriptorWithTextStyle:
                                                        UIFontTextStyleBody];

UIFontDescriptor *boldFontDescriptor =
  [fontDescriptor fontDescriptorWithSymbolicTraits:
                                                  UIFontDescriptorTraitBold];

UIFont *boldFont = [UIFont fontWithDescriptor: boldFontDescriptor size: 0.0];
```

Descriptor Matching

Text style emboldening

```
UIFontDescriptor *fontDescriptor =
  [UIFontDescriptor preferredFontDescriptorWithTextStyle:
                                                        UIFontTextStyleBody];
```

```
UIFontDescriptor *boldFontDescriptor =
  [fontDescriptor fontDescriptorWithSymbolicTraits:
                                                  UIFontDescriptorTraitBold];

UIFont *boldFont = [UIFont fontWithDescriptor: boldFontDescriptor size: 0.0];
```

Descriptor Matching

Text style emboldening

```
UIFontDescriptor *fontDescriptor =
  [UIFontDescriptor preferredFontDescriptorWithTextStyle:
                                                        UIFontTextStyleBody];

UIFontDescriptor *boldFontDescriptor =
  [fontDescriptor fontDescriptorWithSymbolicTraits:
                                                  UIFontDescriptorTraitBold];

UIFont *boldFont = [UIFont fontWithDescriptor: boldFontDescriptor size: 0.0];
```

Descriptor Matching

Text style emboldening

```
UIFontDescriptor *fontDescriptor =
  [UIFontDescriptor preferredFontDescriptorWithTextStyle:
                                                     UIFontTextStyleBody];

UIFontDescriptor *boldFontDescriptor =
  [fontDescriptor fontDescriptorWithSymbolicTraits:
                                                UIFontDescriptorTraitBold];

UIFont *boldFont = [UIFont fontWithDescriptor: boldFontDescriptor size: 0.0];
```

Descriptor Matching

Text style emboldening

```
UIFontDescriptor *fontDescriptor =
  [UIFontDescriptor preferredFontDescriptorWithTextStyle:
                                                        UIFontTextStyleBody];

UIFontDescriptor *boldFontDescriptor =
  [fontDescriptor fontDescriptorWithSymbolicTraits:
                                                  UIFontDescriptorTraitBold];

UIFont *boldFont = [UIFont fontWithDescriptor: boldFontDescriptor size: 0.0];
```

Descriptor Serialization

- `NSCoding` protocol
- Manual serialization
  - `–[UIFontDescriptor fontAttributes]`

Modifying a Font Instance

- Activate typographic features

```
UIFontDescriptorFeatureSettingsAttribute
```

- Remove unwanted characters

```
UIFontDescriptorCharacterSetAttribute
```

Modifying a Font Instance

Time appearance

11:15

vs.

11:15

Modifying a Font Instance

Time appearance

```
NSArray *timeFeatureSettings = @[
  @{
    UIFontFeatureTypeIdentifierKey: @(kNumberSpacingType),
    UIFontFeatureSelectorIdentifierKey: @(kProportionalNumbersSelector)
  },
  @{
    UIFontFeatureTypeIdentifierKey: @(kCharacterAlternativesType),
    UIFontFeatureSelectorIdentifierKey: @(1)
  }];

UIFont *font = …;
UIFontDescriptor *originalDescriptor = [font fontDescriptor];
UIFontDescriptor *timeDescriptor = [originalDescriptor
  fontDescriptorByAddingAttributes: @{
    UIFontDescriptorFeatureSettingsAttribute: timeFeatureSettings }];
UIFont *timeFont = [UIFont fontWithDescriptor: timeDescriptor size: 0.0];
```

Modifying a Font Instance

Time appearance

```
NSArray *timeFeatureSettings = @[
  @{
    UIFontFeatureTypeIdentifierKey: @(kNumberSpacingType),
    UIFontFeatureSelectorIdentifierKey: @(kProportionalNumbersSelector)
  },
  @{
    UIFontFeatureTypeIdentifierKey: @(kCharacterAlternativesType),
    UIFontFeatureSelectorIdentifierKey: @(1)
  }];
```

```
UIFont *font = …;
UIFontDescriptor *originalDescriptor = [font fontDescriptor];
UIFontDescriptor *timeDescriptor = [originalDescriptor
  fontDescriptorByAddingAttributes: @{
    UIFontDescriptorFeatureSettingsAttribute: timeFeatureSettings }];
UIFont *timeFont = [UIFont fontWithDescriptor: timeDescriptor size: 0.0];
```

Modifying a Font Instance

Time appearance

```
NSArray *timeFeatureSettings = @[
  @{
    UIFontFeatureTypeIdentifierKey: @(kNumberSpacingType),
    UIFontFeatureSelectorIdentifierKey: @(kProportionalNumbersSelector)
  },
  @{
    UIFontFeatureTypeIdentifierKey: @(kCharacterAlternativesType),
    UIFontFeatureSelectorIdentifierKey: @(1)
  }];
```

```
UIFont *font = …;
UIFontDescriptor *originalDescriptor = [font fontDescriptor];
UIFontDescriptor *timeDescriptor = [originalDescriptor
  fontDescriptorByAddingAttributes: @{
    UIFontDescriptorFeatureSettingsAttribute: timeFeatureSettings }];
UIFont *timeFont = [UIFont fontWithDescriptor: timeDescriptor size: 0.0];
```

Modifying a Font Instance

Time appearance

```
NSArray *timeFeatureSettings = @[
  @{
    UIFontFeatureTypeIdentifierKey: @(kNumberSpacingType),
    UIFontFeatureSelectorIdentifierKey: @(kProportionalNumbersSelector)
  },
  @{
    UIFontFeatureTypeIdentifierKey: @(kCharacterAlternativesType),
    UIFontFeatureSelectorIdentifierKey: @(1)
  }];

UIFont *font = …;
UIFontDescriptor *originalDescriptor = [font fontDescriptor];
UIFontDescriptor *timeDescriptor = [originalDescriptor
  fontDescriptorByAddingAttributes: @{
    UIFontDescriptorFeatureSettingsAttribute: timeFeatureSettings }];
UIFont *timeFont = [UIFont fontWithDescriptor: timeDescriptor size: 0.0];
```

Modifying a Font Instance

Overly happy snowman

```
This ☃ isn’t my favorite
```

vs.

```
This ⛄ is more sedate
```

Modifying a Font Instance

Overly happy snowman

```
UIFont *font = …;
UIFontDescriptor *originalDescriptor = [font fontDescriptor];
NSMutableCharacterSet *cset = [[originalDescriptor
  objectForKey: UIFontDescriptorCharacterSetAttribute] mutableCopy];
[cset removeCharactersInRange: NSMakeRange(0x2603, 1)];

UIFontDescriptor *snowmanRemoved = [originalDescriptor
  fontDescriptorByAddingAttributes: @{
    UIFontDescriptorCharacterSetAttribute: cset }];
font = [UIFont fontWithDescriptor: snowmanRemoved size: 0.0];
```

Modifying a Font Instance

Overly happy snowman

```
UIFont *font = …;
UIFontDescriptor *originalDescriptor = [font fontDescriptor];
NSMutableCharacterSet *cset = [[originalDescriptor
  objectForKey: UIFontDescriptorCharacterSetAttribute] mutableCopy];
[cset removeCharactersInRange: NSMakeRange(0x2603, 1)];
```

```
UIFontDescriptor *snowmanRemoved = [originalDescriptor
  fontDescriptorByAddingAttributes: @{
    UIFontDescriptorCharacterSetAttribute: cset }];
font = [UIFont fontWithDescriptor: snowmanRemoved size: 0.0];
```

Modifying a Font Instance

Overly happy snowman

```
UIFont *font = …;
UIFontDescriptor *originalDescriptor = [font fontDescriptor];
NSMutableCharacterSet *cset = [[originalDescriptor
  objectForKey: UIFontDescriptorCharacterSetAttribute] mutableCopy];
[cset removeCharactersInRange: NSMakeRange(0x2603, 1)];
```

```
UIFontDescriptor *snowmanRemoved = [originalDescriptor
  fontDescriptorByAddingAttributes: @{
    UIFontDescriptorCharacterSetAttribute: cset }];
font = [UIFont fontWithDescriptor: snowmanRemoved size: 0.0];
```

Modifying a Font Instance

Overly happy snowman

```
UIFont *font = …;
UIFontDescriptor *originalDescriptor = [font fontDescriptor];
NSMutableCharacterSet *cset = [[originalDescriptor
  objectForKey: UIFontDescriptorCharacterSetAttribute] mutableCopy];
[cset removeCharactersInRange: NSMakeRange(0x2603, 1)];

UIFontDescriptor *snowmanRemoved = [originalDescriptor
  fontDescriptorByAddingAttributes: @{
    UIFontDescriptorCharacterSetAttribute: cset }];
font = [UIFont fontWithDescriptor: snowmanRemoved size: 0.0];
```

Adding Fonts

BYOF

- Containers
  - TTF, TTC
  - OTF, OTC
- Formats
  - Apple Advanced Typography (AAT)
    - iOS: ‘morx’ only (no ‘mort’)
  - OpenType
    - Complex scripts (pre-Vista Indic)

Adding Fonts

- Include in app bundle
  - Automatically added using `UIAppFonts` Info.plist key
- Manually add
  - Registered or unregistered

Adding Fonts

Registered

- Referenced by name—same as with installed fonts

```
CTFontManagerRegisterFontsForURL()
CTFontManagerRegisterFontsForURLs()
CTFontManagerRegisterGraphicsFont()
```

Adding Fonts

Unregistered

- Referenced by descriptor—use it or lose it
  - Cannot be matched by name

```
CTFontManagerCreateFontDescriptorsFromURL()
CTFontManagerCreateFontDescriptorFromData()
```

Extra Credit

- Downloadable fonts
    - All OS X fonts
    - Additional international fonts

```
CTFontDescriptorCreateMatchingFontDescriptorsWithProgressHandler()
  + kCTFontDownloadableAttribute
```

- Profiles

```
kCTFontManagerRegisteredFontsChangedNotification
```

Language Tagging

kCTLanguageAttributeName

- Override system language for localized behaviors
  - Glyph substitution
  - Line breaking
  - Font fallbacks

kCTLanguageAttributeName

Glyph substitution

```
[[NSAttributedString alloc] initWithString: @"\u015F\u0163" attributes: @{
  NSFontAttributeName: @"Verdana",
  (id)kCTLanguageAttributeName:
```

@"en"

şţ

@"ro"

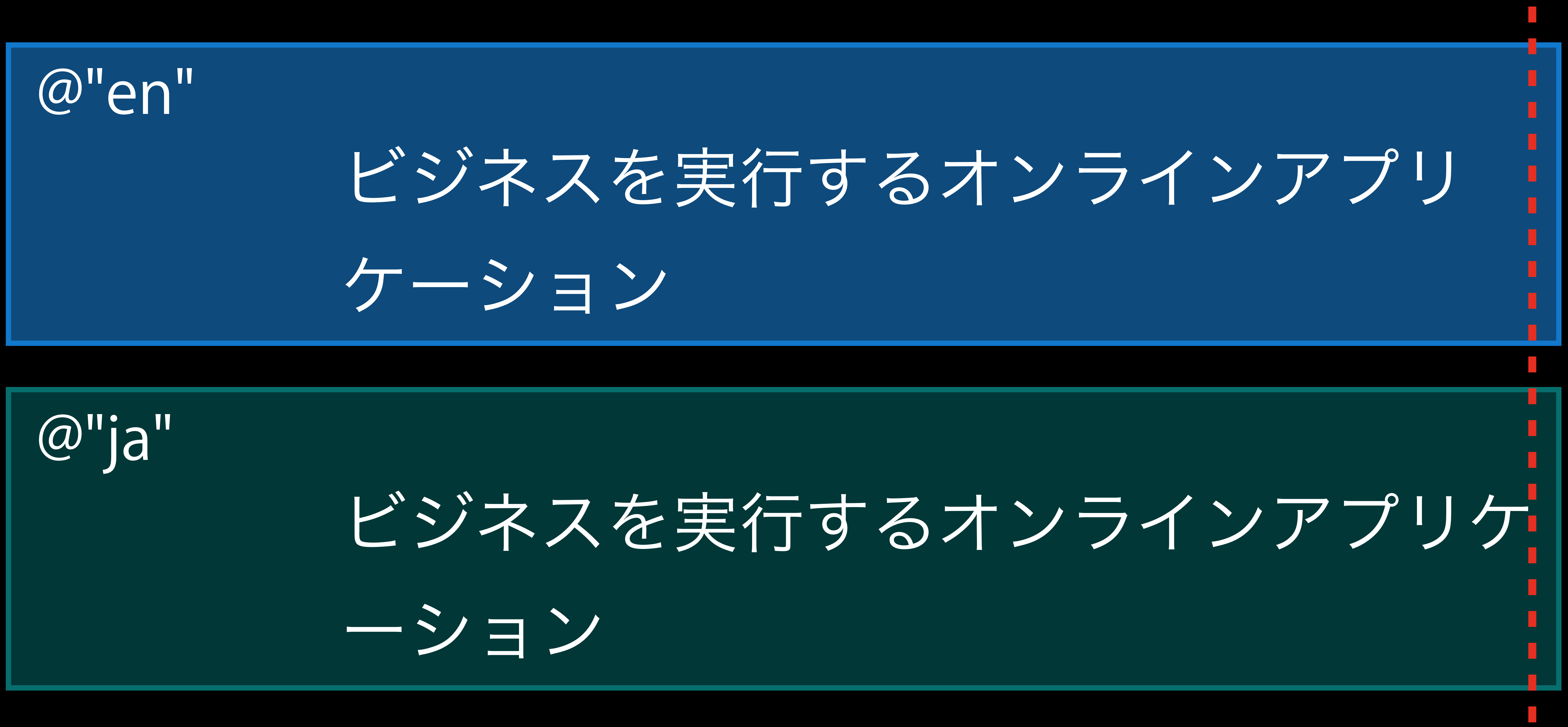
șț

```
}];
```

kCTLanguageAttributeName

Line breaking

```
[[NSAttributedString alloc] initWithString: @"…" attributes: @{
  (id)kCTLanguageAttributeName:
```



@"en"

ビジネスを実行するオンラインアプリ
ケーション

@"ja"

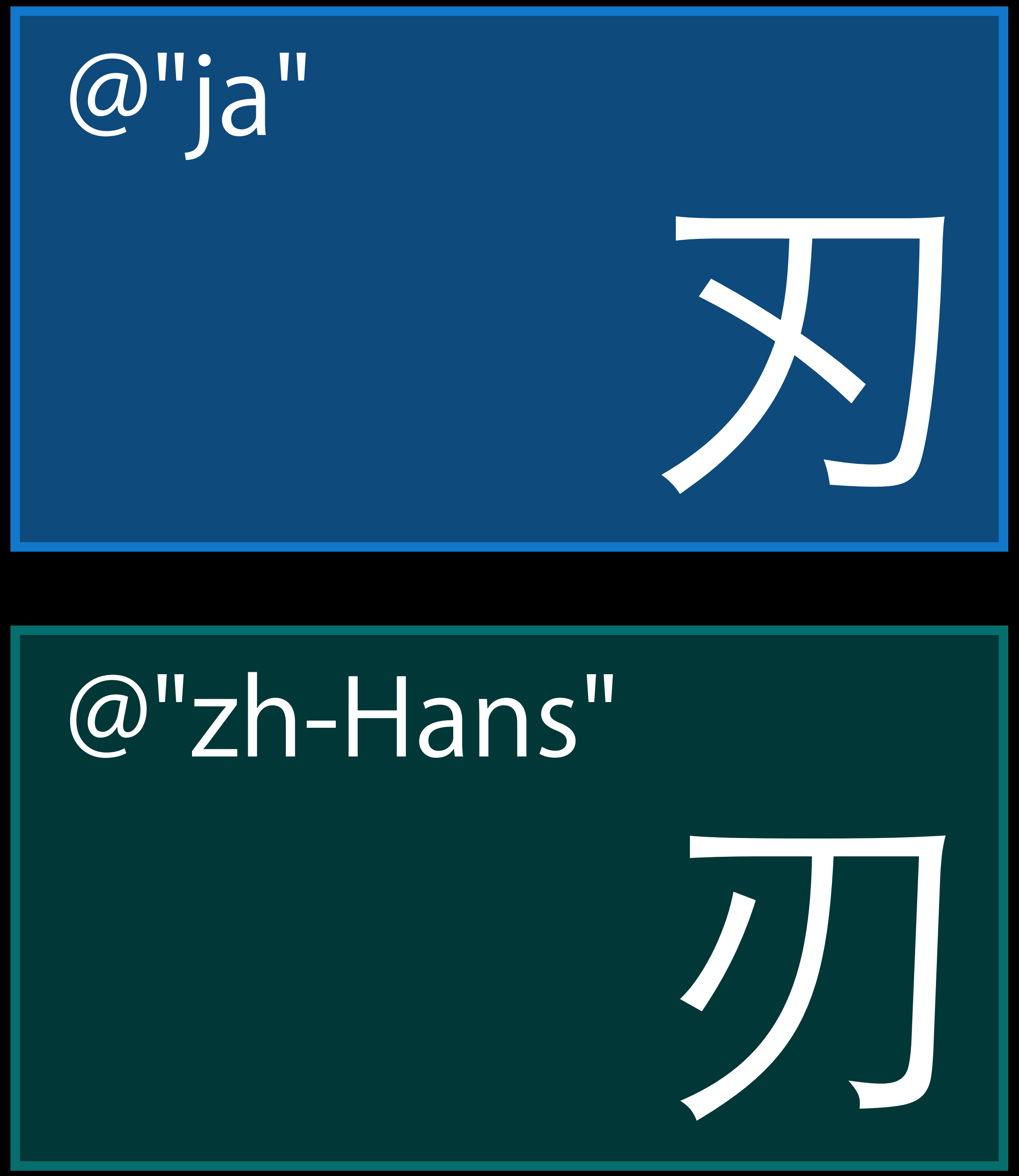
ビジネスを実行するオンラインアプリケ
ーション

```
}];
```

kCTLanguageAttributeName

Font fallbacks

```
[[NSAttributedString alloc] initWithString: @"\u5203" attributes: @{
  (id)kCTLanguageAttributeName:
```



```
}];
```

Modifying Fallbacks

Custom Cascade List

- `UIFontDescriptorCascadeListAttribute`
- Inserted before standard fallbacks

Custom Cascade List

Specifying an Arabic fallback font

apple: التّفّاحة

vs.

apple: التّفّاحة

Custom Cascade List

Specifying an Arabic fallback font

```
UIFontDescriptor *arabic = …;
NSArray *cascadeList = @[ arabic ];

UIFont *font = …;
UIFontDescriptor *originalDescriptor = [font fontDescriptor];
UIFontDescriptor *cascadeListAdded = [originalDescriptor
  fontDescriptorByAddingAttributes: @{
    UIFontDescriptorCascadeListAttribute: cascadeList }];
font = [UIFont fontWithDescriptor: cascadeListAdded size: 0.0];
```

Custom Cascade List

Specifying an Arabic fallback font

```
UIFontDescriptor *arabic = …;
NSArray *cascadeList = @[ arabic ];
```

```
UIFont *font = …;
UIFontDescriptor *originalDescriptor = [font fontDescriptor];
UIFontDescriptor *cascadeListAdded = [originalDescriptor
  fontDescriptorByAddingAttributes: @{
    UIFontDescriptorCascadeListAttribute: cascadeList }];
font = [UIFont fontWithDescriptor: cascadeListAdded size: 0.0];
```

Custom Cascade List

Specifying an Arabic fallback font

```
UIFontDescriptor *arabic = …;
NSArray *cascadeList = @[ arabic ];
```

```
UIFont *font = …;
UIFontDescriptor *originalDescriptor = [font fontDescriptor];
UIFontDescriptor *cascadeListAdded = [originalDescriptor
  fontDescriptorByAddingAttributes: @{
    UIFontDescriptorCascadeListAttribute: cascadeList }];
font = [UIFont fontWithDescriptor: cascadeListAdded size: 0.0];
```

Custom Cascade List

Specifying an Arabic fallback font

```
UIFontDescriptor *arabic = …;
NSArray *cascadeList = @[ arabic ];

UIFont *font = …;
UIFontDescriptor *originalDescriptor = [font fontDescriptor];
UIFontDescriptor *cascadeListAdded = [originalDescriptor
  fontDescriptorByAddingAttributes: @{
    UIFontDescriptorCascadeListAttribute: cascadeList }];
font = [UIFont fontWithDescriptor: cascadeListAdded size: 0.0];
```

More Information

Jake Behrens
App Frameworks Evangelist
behrens@apple.com

Documentation
http://developer.apple.com

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

Related Sessions

Introduction to Text Kit	Presidio Wednesday 2:00PM
Advanced Text Layouts and Effects with Text Kit	Mission Thursday 2:00PM

Summary

- Text styles on iOS 7
  - Use Auto Layout or otherwise adapt to font and size changes
- Language tagging
  - Override system language for known content
- Modifying fallbacks
  - Insert descriptors prior to default fallbacks

WWDC2013