iOS 11.0

This article summarizes the key developer-related features introduced in iOS 11, which runs on currently shipping iOS devices. The article also lists the documents that describe new features in more detail.

For late-breaking news and information about known issues, see *iOS 11 Release Notes*.

For a complete list of new, modified, and deprecated APIs, see Apple Developer Documentation.

To learn about new features for SafariServices and WebKit, see *What's New in Safari*.

To learn about what's new in Swift, see Swift Language in *What's New in Xcode* and *The Swift Programming Language (Swift 4)*.

General

- **New in iOS 11.0 – Support for binary (nontext) barcodes.**
  - Added APIs to AV Foundation, Core Image, and SiriKit to support detection, decoding, and creation of barcodes with binary content.
  - Added a new barcode descriptor object, `CIBarcodeDescriptor` to Core Image to provide interoperability with AV Foundation and the Vision APIs.
- **New in iOS 11.0 – MusicKit.**
  - MusicKit gives your app access to the full Apple Music catalog, and to the user's library.
  - Added and updated functionality in StoreKit for retrieving client tokens and storefront identifiers.
  - Added and updated functionality in Media Player to enable station playback.
  - Added display of custom messages in the Apple Music subscriber setup flow.

App Frameworks

- **New in iOS 11.0 – Support for drag and drop.** Drag and drop in iOS lets a user drag items from one location to another onscreen, either within a single app or in different apps.
  - Added functionally to identify views as drag sources and as drag destinations.
  - Added customizable previews and set-down animations.

  See Drag and Drop in the documentation.
- **New in iOS 11.0 – Browsing local and iCloud documents.**

- Added view controllers for browsing documents stored locally and in the cloud. See `UIDocumentBrowserViewController` and `UIDocumentBrowserTransitionController`.
- Added `UIDocumentBrowserAction`, an object for creating a custom action for the document browser.
- Added the FileProvider and FileProviderUI frameworks for adding third party storage services.

See Adding a Document Browser to Your App.

- Improved Dynamic Type support.
  - Added `UIFontMetrics`, an object for creating custom fonts that scale based on the currently selected text size.
  - Updated Auto Layout to dynamically adjust spacing based on the font sizes when a baseline anchor is part of a constraint that uses the system spacing.
  - Added an attribute to preserve vector data for PDF assets to enable smooth scaling. You can use this attribute to show a larger version of bar items and segmented control items, as well as to adjust image sizes to match the user's text size. To enable scaling in the PDF, in the asset catalog Attributes inspector, select "Preserve vector data" for the PDF to enable scaling.
  - Added `UIAccessibilityContentSizeCategoryImageAdjusting`, a protocol for scaling images for accessibility text sizes.
- Improved Auto Layout support for Dynamic Type.
  - Updated `NSLayoutXAxisAnchor` and `NSLayoutYAxisAnchor` to provide factory methods that create constraints using the system spacing between two anchors. Previously the only way to create such a constraint was with the dash (`-`) in the Visual Format Language.
  - Added an option to `NSLayoutFormatOptions` for creating a Visual Format Language string that uses baseline-to-baseline spacing.
  - Updated `UIStackView` to enable system spacing and customized spacing.
- Updated text content to work with App Password autofill.
  - Added `username` and `password` properties to `UITextContentType`.
- Added `primaryEdge`, a property of `UISplitViewController` for setting the side for the master view controller.
- Added `sectionInsetReference`, a new enum property for `UICollectionViewFlowLayout`, that specifies the boundary used for relative section insets.
- Updated keyboard extensions.
  - Added `selectedText`, a property of `UITextDocumentProxy` that returns the currently selected text in the document.
  - Added `documentIdentifier`, a property of `UITextDocumentProxy` that specifies whether the user is navigating to a new text widget.
  - Added `hasFullAccess`, a property of `UIInputViewController` that checks keyboard permissions.
  - Added `needsInputModeSwitchKey`, a property of `UIInputViewController` to control the display of the input-mode switch key.
  - Added new system permissions in Settings for app access to included keyboard extensions.
- Improved API for available storage space.

- Added new keys to the `URL` class for different usage scenarios.
    - The `volumeAvailableCapacityForImportantUsageKey` key returns total amount of bytes available for operations explicitly requested by user or essential to proper functioning of your apps.
    - The `volumeAvailableCapacityForOpportunisticUsageKey` key returns total amount of bytes available for storing nonessential items, such as content predownloaded for performance that may or may not get used by the user.

Graphics and Games

- **New in iOS 11.0 – Support for augmented reality.**
    - Added the ARKit framework that combines device motion tracking, camera scene capture, advanced scene processing, and display conveniences to simplify the task of building an AR experience.
- **New in iOS 11.0 – High performance image analysis.**
    - Added the Vision framework for detecting faces, bar codes, text, image horizon, and rectangular regions.
    - Provided support for integrating the Vision framework with Core ML to run custom models on images.
    - Added object-tracking in video.
    - Added support for image registration.
- **New in iOS 11.0 – Ability to write custom image blending kernels for Core Image.**
    - Added `CIBlendKernel`, a special type of `CIColorKernel` to blend two images (supported by `CIRenderDestination` and `CIImageAccumulator`).
    - Added `init(functionName:fromMetalLibraryData:)` to `CIKernel` for writing kernels using Metal to benefit from the improved language features and the reduced compile time.
- **New in iOS 11.0 – Lightweight render destination.**
    - Added `CIRenderDestination`, an object for creating renderers that return to the caller after the work has been issued. You can specify all the destination attributes of the renderer for different destinations, including a surface (`IOSurface`), Core Video pixel buffer (`CVPixelBuffer`), GL textures, Metal textures, and memory.
- Extended the ReplayKit framework.
    - Updated `RPScreenRecorder` for screen capture and back camera support.
- Added new Core Image filters `CITextImageGenerator`, `CIColorCurves`, `CILabDeltaE`, `CIBokehBlur`, `CIMinMaxRed`, and `CIBicubicScaleTransform`.

Metal 2

Metal 2 contains significant additions and updates to Metal, the Metal Shading Language, and the Metal Performance Shaders framework. Items below indicate where the updates occur:

- MTL: An update in the Metal framework.
- MSL: An update in the Metal Shading Language.
- MPS: An update in the Metal Performance Shaders framework.

- **MPS: New in Metal 2 – Cross-platform Metal Performance Shaders support.**
  - All Metal Performance Shaders functionality is available in iOS 11.0, tvOS 11.0, and macOS 10.13.
- **MPS: New in iOS 11.0 – Neural network support.**
  - Added support for neural networks to the Metal Performance Shaders framework.
  - Added graphs to offer a higher level API for simplifying the creation of neural networks, including objects that allow state to be transferred between nodes in a neural network.
  - Added convolutional neural networks (CNN) to support implementing and running deep learning using previously obtained training data.
  - Added recurrent neural networks for implementing inference on images and matrices.
- **New in iOS 11.0 – Argument buffers.** Group your resources into an argument buffer (AB) to reduce CPU overhead.
  - MSL: Added the `[[id(n)]]` attribute qualifier to identify resources in an AB structure.
  - MTL: Added the `MTLArgumentEncoder` protocol to encode resources into an AB.
- **MTL: New in iOS 11.0 – Programmable sample positions.** Configure the position of samples when rendering to a multisampled render target.
  - Updated the `MTLRenderPassDescriptor` class to set and get sample positions for a render pass.
- **MSL: New in iOS 11.0 – Uniform type.**
  - Added the `uniform` type to declare variables that are uniform for all threads that execute the graphics or compute function of a draw or dispatch call.
- **MSL: New in iOS 11.0 – Array of samplers.**
  - Added the `array<sampler, size_t N>` type to store an array of samplers.
- **MTL: New in iOS 11.0 – BGR10A2 pixel format.**
  - Added the `bgr10A2Unorm` pixel format to present wide color content on P3 displays.
- MPS: Added new filters.
  - Added filters for image statistics, such as computing the mean and variance for an image region.
  - Added filters for combining two images together, such as an element-wise sum.

    - Added filters for matrix decomposition and solving, such as decomposition using Cholesky or LU (lower upper) factorization.

- MSL: Extended function specialization. Members of a structure used in a graphics, compute, or user function can be used with function constants.
    - Extended `[[color(n)]]` and `[[raster_order_group(index)]]` attribute qualifiers to work with function constants.
- MTL: Extended vertex formats.
    - Added new `MTLVertexFormat` values for small formats such as `char`, `short`, and `half`.
- Added dual-source blending support to iOS. Output two source colors to a single render target in a fixed-function blending operation.
    - MSL: Added a new `[[index(i)]]` attribute qualifier to the `[[color(n)]]` attribute qualifier to output a second source color.
    - MTL: Updated `MTLBlendFactor` to operate on a second source color.

App Services

- **New in iOS 11.0 – Support for machine learning models.**
    - Added the Core ML framework for easily integrating machine learning models into apps.
- **New in iOS 11.0 – SiriKit support for visual codes.**
    - Added the Visual Codes domain to Sirkit to support showing visual codes for exchanging payment and contact information.
- **New in iOS 11.0 – SiriKit support for notes and to-do lists.**
    - Added the Lists and Notes domain to SiriKit to support using Siri to add notes, interact with to-do lists, and interact with reminders.
- Added intents to SiriKit domains.
    - Added ride canceling and feedback to the Ride Booking domain.
    - Added transferring money and searching for accounts to the Payments domain.
- **New in iOS 11.0 – Find the heading of the device.**
    - Added `heading` to `CMDeviceMotion`, a property that returns the heading angle with respect to the `CMAttitudeReferenceFrame`. The returned value is the heading in degrees as a `double`. A negative value is returned when the reference frame is `xArbitraryZVertical` or `xArbitraryCorrectedZVertical`.
- **New in iOS 11.0 – Multipath TCP.**
    - Added support for using multiple interfaces, like Wi-Fi and cellular, to transmit a single data stream, by extending `URLSessionConfiguration` to support Multipath TCP as defined in IETF RFC 6824. See `URLSessionConfiguration.MultipathServiceType`.

- **New in iOS 11.0 - DNS Proxy.**
  - Added a new DNS Proxy app extension type to the Network Extension framework.
- Enhanced end user transaction flow in Apple Pay.
  - Added `PKPaymentError` to PassKit, a structure for detailed reporting of errors in a user's shipping and payment information, and for authorization errors. Developers can use the information to provide a customized error string.
  - Updated the handler methods in `PKPaymentAuthorizationControllerDelegate` to receive a `PKPaymentError`.
  - Updated `PKPaymentRequest` to use `PKContactField` for contact information.
  - Added `supportedCountries` to `PKPaymentRequest` for specifying supported countries for a transaction.
  - Added support for presenting payment buttons even if there are no supported payment methods in Wallet. Apple Pay now handles payment without leaving your app, and then returns to checkout.
- **New in iOS 11.0 - Promoting in-app purchases on the App Store.**
  - Developers can promote up to 20 in-app purchases on their App Store product page. Customers can start their purchase on the App Store, and then be taken to the app to complete the transaction.
  - Added `paymentQueue(_:shouldAddStore:for:)`, a new method of `SKPaymentTransactionObserver` for promoted in-app purchases. Apps need to support this delegate for promoted in-app purchases to display on the App Store.
- **New in iOS 11.0 - Live messages.**
  - Added `MSMessageLiveLayout`, a new message layout for showing live messages that can display dynamic content, such as games. Each live message has its own `MSMessagesAppViewController`, and there can be multiple active live messages on the screen at once.

    The following code shows adding a live message to a message stream, including an alternate layout for devices that do not support live messages.

```
guard let conversation = activeConversation else {

       fatalError("No active conversation")

}


let alternateLayout = MSMessageTemplateLayout()

alternateLayout.image = UIImage(named: "SuperSweetGameImage")

alternateLayout.caption = "$(\(conversation.localParticipantIdentifier)) wants
to play a game!"

let layout = MSMessageLiveLayout(alternateLayout: alternateLayout)


let message = MSMessage()
```

```
message.layout = layout


conversation.insert(message, completionHandler: nil)
```

- Enhanced triggers for HomeKit.
  - Enhanced time-based conditions for triggers. `HMSignificantTimeEvent` specifies an offset from sunrise and sunset. `HMCalendarEvent` specifies a date and time. `HMDurationEvent` specifies a time interval.
  - Added `HMCharacteristicThresholdRangeEvent` to support tracking the state of an accessory within a range, such as running an action when the temperature is between 68 and 72 degrees.
  - Added `HMPresenceEvent` for adding a condition based on the presence or absence of users.
  - Updated `HMEventTrigger` to enable multiple recurrences of the event.
- Added `home:didUpdateHomeHubState:` to support receiving updates of the home hub state.
- Updated MapKit for clearer display of developer data.
  - Added `mutedStandard`, a new map display mode that emphasizes developer data.
  - Added properties to customize how annotations behave when collisions occur. Developers use a combination of `displayPriority`, `collisionMode`, and `clusteringIdentifier` to influence which annotations remain on the map.
- Added the `authorizationStatus` method to the `CMAltimiter`, `CMPedometer`, `CMMotionActivityManager`, and `CMSensorRecorder` classes of the Core Motion framework. The method is used to determine if an app is authorized to recieve data from a source.

Media and Web

- **New in iOS 11.0 – Support for High Efficiency Video Coding (HEVC).** High Efficiency Video Coding (HEVC) is a new standard for video encoding that offers substantially better compression than H.264 at the same level of visual quality.
  - Added support for using AV Foundation to play back movies containing HEVC-encoded tracks, and to capture and export videos.
  - Added support for using `VideoToolbox` clients to encode and decode HEVC video bitstreams.
- **New in iOS 11.0 – Support for High Efficiency Image Format (HEIF).** High Efficiency Image Format (HEIF) is a new standard of image compression that nearly doubles current data compression ratios for the same level of image quality.
  - Added functionality to the Photos and Core Image frameworks to display, encode, and export HEIF images.
- **New in iOS 11.0 – Support for capturing and manipulating depth data, and enhanced photo capture.**

    - Added objects to AV Foundation for capturing and representing depth data. See `AVCaptureDepthDataOutput`, `AVDepthData`, and related APIs.
    - Added `AVCapturePhoto`, an object that encapsulates the information for a captured photo and supports HEVC and HEIC encoded images.
    - Updated `AVCapturePhotoOutput` to provide more information about camera features and supported output formats.

- **New in iOS 11.0 – Automatic storage management.**
    - Added automatic storage management of HTTP live streaming assets to `AVAssetDownloadTask`. The system can automatically purge expired or unneeded downloads when space is required. Use priorities to influence the purging policy.

- **New in iOS 11.0 – AirPlay 2.**
    - Improved AirPlay reliability for some audio playback interfaces in AV Foundation. To take advantage of the increased reliability, play audio using `AVPlayer` or the new `AVSampleBufferAudioRenderer` object.
    - Added multiple speaker support to AirPlay for long-form audio, such as music and podcasts. To mark your application as presenting long-form audio, invoke the `AVAudioSession` method `setCategory(_:mode:routeSharingPolicy:options:)` and use `AVAudioSessionRouteSharingPolicyLongForm` as the parameter value.
    - Added `AVRoutePickerView` to the AVKit framework and `AVRouteDetector` to the AVFoundation framework for enabling users to choose the route for playing content when multiple routes are available. Use `AVRouteDetector` to determine if multiple routes are available when route detection is enabled. If multiple routes are available, use `AVRoutePickerView` to present an interface for the user to choose the routes.

- Added FairPlay streaming key management.
    - Improved the functionality of `AVContentKeySession`. Use `AVContentKeySession` to initiate content key requests independent of playback or downloading of media assets. Objects conforming to the `AVContentKeyRecipient` protocol, such as `AVURLAsset`, can be added as a recipient to `AVContentKeySession` to obtain access to existing content keys and initiate new content key requests

- Added more Live Photo adjustments.
    - Added a collection of Live Photo adjustments, called *effects*, that render the live photo as Loop, Bounce, or Long Exposure. Unlike regular live photos, Loop and Bounce videos will play in a continuous loop.
    - Added a `playbackStyle`, a new property that identifies how to present the `PHAsset` to the user.

System

- **New in iOS 11.0 – Hotspot configuration.**
    - Added a network extension for hotspot configuration. See `NEHotspotConfiguration`.

- **New in iOS 11.0 - Detect NFC tags and read messages that contain NDEF data.**
    - Added Core NFC, a new framework for reading Near Field Communications (NFC) tags and data in NFC Data Exchange Format (NDEF.)
- Updated the Core Bluetooth framework.
    - Added support for L2CAP Channels.
    - Extended session restoration to work across Bluetooth resets and device reboots.
    - Updated the APIs in the Core Bluetooth framework to match across iOS, tvOS, watchOS, and macOS, and marked the platform availability of each API.
- APFS is now the default filesystem.
    - Added normalization-insensitive support for a case sensitive filesystem.