

[ARKit / ARKit in visionOS](#)[API Collection](#)

ARKit in visionOS

Create immersive augmented reality experiences.

Overview

ARKit in visionOS offers a new set of sensing capabilities that you adopt individually in your app, using data providers to deliver updates asynchronously. The available capabilities include:

- **Plane detection.** Detect surfaces in a person's surroundings and use them to anchor content.
- **World tracking.** Determine the position and orientation of Apple Vision Pro relative to its surroundings, and add world anchors to place content.
- **Hand tracking.** Use a person's hand and finger positions as input for custom gestures and interactivity.
- **Scene reconstruction.** Build a mesh of a person's physical surroundings and incorporate it into your immersive spaces to support interactions.
- **Image tracking.** Look for known images in a person's surroundings and use them as anchor points for custom content.
- **Object tracking.** Use 3D reference objects to find and track real-world objects in a person's environment.
- **Barcode detection.** Detect and scan QR codes and barcodes in a variety of formats in a person's surroundings.
- **Room tracking.** Use room anchors to identify specific rooms and implement per-room experiences.
- **Light estimation.** Understand the lighting characteristics of a room to help improve the appearance of shiny or semi-reflective materials in your virtual content.
- **Camera frames.** Access camera frames from a device in several formats.

- **Accessory tracking.** Work with the real-time position and orientation of accessories that a person is using.



Topics

Setup

📄 Setting up access to ARKit data

Check whether your app can use ARKit and respect people's privacy.

`class ARKitSession`

The main entry point for receiving data from ARKit.

`protocol DataProvider`

A source of live data from ARKit.

`enum DataProviderState`

The possible states of a data provider.

`protocol Anchor`

The identity, location, and orientation of an object in world space.

`protocol TrackableAnchor`

An anchor that can gain and lose its tracking state over the course of a session.

`struct ARKitCoordinateSpace`

An object which represents an ARKit coordinate space.

Barcode detection

`class BarcodeDetectionProvider`

An object that provides the real-time position of barcodes the framework detects in a person's environment.

`struct BarcodeAnchor`

A barcode's position in a person's surroundings.

Camera sampling

`class CameraFrameProvider`

An object that provides camera streams.

`struct CameraFrame`

The representation of a camera frame.

`struct CameraVideoFormat`

A structure that represents a camera video format.

Rendering

`class StereoPropertiesProvider`

The StereoPropertiesProvider serves the latest viewpoint properties on the device.

`struct ViewpointProperties`

The ViewpointProperties is a record of render camera transforms at some particular time.

Camera region

`class CameraRegionProvider`

A camera region provider. An enterprise license is required to use the CameraRegionProvider. The provider will not deliver any data without it. The app must include the following entitlement: com.apple.developer.arkit.camera-region.allow

```
struct CameraRegionAnchor
```

Represents a region in space to capture a camera stream of.

Plane detection

{} Placing content on detected planes

Detect horizontal surfaces like tables and floors, as well as vertical planes like walls and doors.

```
class PlaneDetectionProvider
```

A source of live data about planes in a person's surroundings.

```
struct PlaneAnchor
```

An anchor that represents horizontal and vertical planes.

World tracking

{} Tracking specific points in world space

Retrieve the position and orientation of anchors your app stores in ARKit.

```
class WorldTrackingProvider
```

A source of live data about the device pose and anchors in a person's surroundings.

```
struct WorldAnchor
```

A fixed location in a person's surroundings.

```
struct DeviceAnchor
```

The position and orientation of Apple Vision Pro.

Hand tracking

{} Happy Beam

Leverage a Full Space to create a fun game using ARKit.

```
class HandTrackingProvider
```

A source of live data about the position of a person's hands and hand joints.

```
struct HandAnchor
```

A hand's position in a person's surroundings.

```
struct HandSkeleton
```

A collection of joints in a hand.

Scene reconstruction

{ } Incorporating real-world surroundings in an immersive experience

Create an immersive experience by making your app's content respond to the local shape of the world.

```
class SceneReconstructionProvider
```

A source of live data about the shape of a person's surroundings.

```
struct MeshAnchor
```

A volume of space that contains a mesh of a person's surroundings.

Image tracking

{ } Tracking and altering images

Create images from rectangular shapes found in the user's environment, and augment their appearance.

{ } Detecting Images in an AR Experience

React to known 2D images in the user's environment, and use their positions to place AR content.

 Tracking preregistered images in 3D space

Place content based on the current position of a known image in a person's surroundings.

```
class ImageTrackingProvider
```

A source of live data about a 2D image's position in a person's surroundings.

```
struct ImageAnchor
```

A 2D image's position in a person's surroundings.

```
struct ReferenceImage
```

A 2D image the system uses as a reference to find the same image in a person's surroundings.

Geometry

```
struct GeometryElement
```

A container for vertex indices of lines or triangles.

```
struct GeometrySource
```

A container for geometrical vector data.

Lighting estimation

```
class EnvironmentLightEstimationProvider
```

A source of live data about lighting information in the environment.

```
struct EnvironmentProbeAnchor
```

An environment probe in the world.

Object tracking

```
class ObjectTrackingProvider
```

A source of real-time position of reference objects in a person's environment.

```
struct ObjectAnchor
```

A reference object ARKit is tracking.

{} Exploring object tracking with ARKit

Find and track real-world objects in visionOS using reference objects trained with Create ML.

📄 Implementing object tracking in your visionOS app

Create engaging interactions by training models to recognize and track real-world objects in your app.

Accessory tracking

```
class AccessoryTrackingProvider
```

Provides the real time position of accessories in the user's environment.

```
struct Accessory
```

Represents an accessory to be tracked.

```
struct AccessoryAnchor
```

Represents a tracked accessory.

```
{} Tracking accessories in volumetric windows
```

Translate the position and velocity of tracked handheld accessories to throw virtual balls at a stack of cans.

```
{} Tracking a handheld accessory as a virtual sculpting tool
```

Use a tracked accessory with Apple Vision Pro to create a virtual sculpture.

Room tracking

```
class RoomTrackingProvider
```

A source of real-time information about the room that a person is currently in.

```
struct RoomAnchor
```

The representation of a room ARKit is currently tracking.

```
enum SurfaceClassification
```

A value describing the classification of a surface.

```
{} Building local experiences with room tracking
```

Use room tracking in visionOS to provide custom interactions with physical spaces.

Shared coordinate spaces

```
class SharedCoordinateSpaceProvider
```

Provides ability to establish a shared coordinate space among multiple participants.

See Also

visionOS

```
📄 Setting up access to ARKit data
```

Check whether your app can use ARKit and respect people's privacy.

```
class ARKitSession
```

The main entry point for receiving data from ARKit.

```
protocol DataProvider
```

A source of live data from ARKit.

```
protocol Anchor
```

The identity, location, and orientation of an object in world space.