

[Speech](#) / AssetInventory

Class

AssetInventory

Manages the assets that are necessary for transcription or other analyses.

iOS 26.0+ | iPadOS 26.0+ | Mac Catalyst 26.0+ | macOS 26.0+ | visionOS 26.0+

```
final class AssetInventory
```

Overview

Before using the [SpeechAnalyzer](#) class, you must install assets required by the modules you plan to use. These assets are machine-learning models downloaded from Apple's servers and managed by the system. Once you download, install, or use an asset, the system retains and updates it automatically, and shares it with other apps. The system makes a certain number of locale-specific asset reservations available to your app to limit storage space and network usage.

Your app does not work with assets directly. Instead, your app configures module objects. The system uses the modules' configuration to determine what assets are relevant.

Install assets

Installing an asset is a four-step process:

1. Create analyzer modules in the configurations that you wish to use. These modules can be discarded when no longer needed; the system installs assets using the modules' configuration, not their object identity.
2. Assign your app's asset reservations to those locales. The class does this automatically if needed, but you can also call [`reserve\(locale:\)`](#) to do this manually. This step is only necessary for modules with locale-specific assets; that is, modules conforming to [LocaleDependentSpeechModule](#). You can skip this step for other modules.

3. Start downloading the required assets for the modules' configuration. Call [assetInstallationRequest\(supporting:\)](#) to obtain an instance of [AssetInstallationRequest](#) and call its [downloadAndInstall\(\)](#) method.
4. Wait for the download to finish. Note that the download may finish immediately; the assets may have already been downloaded if the assets were preinstalled on the system, another app already downloaded them, or a previous module configuration used the same assets.

Once assets are downloaded, they persist between app launches and are shared between apps. The system may unsubscribe your app from assets that haven't been used in a while.

Manage assets

When your app no longer needs assets for a particular locale, call [release\(reservedLocale:\)](#) to free up that reservation. The system will remove the assets at a later time.

Topics

Downloading and installing assets

```
static func assetInstallationRequest(supporting: [any SpeechModule])  
async throws -> AssetInstallationRequest?
```

Returns an installation request object, which is used to initiate the asset download and monitor its progress.

Checking asset status

```
static func status(forModules: [any SpeechModule]) async -> Asset  
Inventory.Status
```

Returns the status for the list of modules.

```
enum Status
```

Type Properties

```
static var maximumReservedLocales: Int
```

The number of locale reservations permitted to an app.

```
static var reservedLocales: [Locale]
```

The app's current asset locale reservations.

Type Methods

```
static func release(reservedLocale: Locale) async -> Bool
```

Removes an asset locale reservation.

```
static func reserve(locale: Locale) async throws -> Bool
```

Add an asset locale to the app's current reservations.

See Also

Essentials

{} Bringing advanced speech-to-text capabilities to your app

Learn how to incorporate live speech-to-text transcription into your app with `SpeechAnalyzer`.

```
actor SpeechAnalyzer
```

Analyzes spoken audio content in various ways and manages the analysis session.