

[Translation](#) / TranslationSession

## Class

# TranslationSession

A class that performs translations between a pair of languages.

iOS 18.0+ | iPadOS 18.0+ | Mac Catalyst 26.0+ | macOS 15.0+

```
class TranslationSession
```

## Overview

This class provides a flexible way for you to translate one or more lines of text at a time. There are two ways in which you can initialize a `TranslationSession`. One way you can obtain an instance of this class is by adding a `.translationTask()` to a SwiftUI view within your app. You can either add a `translationTask(_:_:action:)` or a `translationTask(source:target:action:)` function to the SwiftUI view containing the content you want to translate, like a `Text` view. After adding the task, the function passes you an instance of a translation session in its action closure. With this instance, you can use one or more of the translate functions to translate a single string or multiple strings of text.

Another way for contexts where there's no UI, you can directly initialize the `TranslationSession` using `init(installedSource:target:)` to translate between languages. This initializer requires that you specify which source language you use and throws an error if the languages aren't already installed on the person's device.

The following example demonstrates how to translate a single string of text within a SwiftUI view:

```
struct TranslationExample: View {
    var sourceText: String
    var sourceLanguage: Locale.Language?
    var targetLanguage: Locale.Language?
```

```

@State private var targetText: String?

var body: some View {
    Text(targetText ?? sourceText)
        .translationTask(
            source: sourceLanguage,
            target: targetLanguage
        ) { session in
            do {
                let response = try await session.translate(sourceText)
                targetText = response.targetText
            } catch {
                // Handle error.
            }
        }
}

```

### Note

All translations using the `TranslationSession` class are processed on the user's device. Apple may collect API usage and performance metrics including the app bundle ID and the original and translated language, but this data does not include the original or translated content.

## Topics

### Initializing a translation session

```
convenience init(installedSource: Locale.Language, target: Locale.Language?)
```

Creates a translation session to translate between a given source and target language already installed on device.

### Preparing for translation

```
struct Configuration
```

A type containing the information to use when performing a translation.

```
func prepareTranslation() async throws
```

Asks for permission to download translation languages without doing any translations.

## Getting the language configuration

```
let sourceLanguage: Locale.Language?
```

The input language to translate from.

```
let targetLanguage: Locale.Language?
```

The output language to translate into.

## Translating the text

```
func translate(String) async throws -> TranslationSession.Response
```

Translates a single string of text.

```
func translate(batch: [TranslationSession.Request]) -> Translation  
Session.BatchResponse
```

Translates multiple strings of text of the same language, returning a sequence of responses as they're available.

```
func translations(from: [TranslationSession.Request]) async throws -> [  
TranslationSession.Response]
```

Translates multiple strings of text of the same language, returning the results all at once when complete.

```
struct Request
```

A translation request containing a single item of text to translate.

```
struct Response
```

The response to a translation request.

```
struct BatchResponse
```

A type that provides asynchronous access to translation responses.

## Accessing the session properties

```
var canRequestDownloads: Bool
```

A boolean value that indicates whether a translation session can request language downloads.

```
var isReady: Bool
```

A boolean value that indicates whether the system has installed the source and target languages of the session and is ready to begin translation.

## Cancelling a translation session

```
func cancel()
```

Attempts to stop all ongoing work for the translation session.

---

## See Also

### Essentials

`{}` Translating text within your app

Display simple system translations and create custom translation experiences.

```
nonisolated func translationPresentation(isPresented: Binding<Bool>,
text: String, attachmentAnchor: PopoverAttachmentAnchor =
.rect(.bounds), arrowEdge: Edge = .top, replacementAction: ((String) ->
Void)? = nil) -> some View
```

Presents a translation popover when a given condition is true.

```
nonisolated func translationTask(_ configuration: TranslationSession.
Configuration?, action: @escaping (TranslationSession) async -> Void) -
> some View
```

Adds a task to perform before this view appears or when the translation configuration changes.

```
nonisolated func translationTask(source: Locale.Language? = nil, target
: Locale.Language? = nil, action: @escaping (TranslationSession) async
-> Void) -> some View
```

Adds a task to perform before this view appears or when the specified source or target languages change.