

[Foundation Models](#) / [SystemLanguageModel](#)

Class

SystemLanguageModel

An on-device large language model capable of text generation tasks.

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

```
final class SystemLanguageModel
```

Mentioned in

-  Improving the safety of generative model output
-  Generating content and performing tasks with Foundation Models
-  Loading and using a custom adapter with Foundation Models

Overview

The `SystemLanguageModel` refers to the on-device text foundation model that powers Apple Intelligence. Use `default` to access the base version of the model and perform general-purpose text generation tasks. To access a specialized version of the model, initialize the model with `SystemLanguageModel.UseCase` to perform tasks like `contentTagging`.

Verify the model availability before you use the model. Model availability depends on device factors like:

- The device must support Apple Intelligence.
- Apple Intelligence must be turned on in Settings.

Use `SystemLanguageModel.Availability` to change what your app shows to people based on the availability condition:

```
struct GenerativeView: View {  
    // Create a reference to the system language model.  
    private var model = SystemLanguageModel.default  
  
    var body: some View {  
        switch model.availability {  
            case .available:  
                // Show your intelligence UI.  
            case .unavailable(.deviceNotEligible):  
                // Show an alternative UI.  
            case .unavailable(.appleIntelligenceNotEnabled):  
                // Ask the person to turn on Apple Intelligence.  
            case .unavailable(.modelNotReady):  
                // The model isn't ready because it's downloading or because  
                // of other system reasons.  
            case .unavailable(let other):  
                // The model is unavailable for an unknown reason.  
        }  
    }  
}
```

Topics

Loading the model with a use case

```
convenience init(useCase: SystemLanguageModel.UseCase, guardrails:  
    SystemLanguageModel.Guardrails)
```

Creates a system language model for a specific use case.

```
struct UseCase
```

A type that represents the use case for prompting.

```
struct Guardrails
```

Guardrails flag sensitive content from model input and output.

Loading the model with an adapter

 Loading and using a custom adapter with Foundation Models

Specialize the behavior of the system language model by using a custom adapter you train.

`com.apple.developer.foundation-model-adapter`

A Boolean value that indicates whether the app can enable custom adapters for the Foundation Models framework.

`convenience init(adapter: SystemLanguageModel.Adapter, guardrails: SystemLanguageModel.Guardrails)`

Creates the base version of the model with an adapter.

`struct Adapter`

Specializes the system language model for custom use cases.

Checking model availability

`var isAvailable: Bool`

A convenience getter to check if the system is entirely ready.

`var availability: SystemLanguageModel.Availability`

The availability of the language model.

`enum Availability`

The availability status for a specific system language model.

Retrieving the supported languages

`var supportedLanguages: Set<Locale.Language>`

Languages that the model supports.

Getting the default model

```
static let `default`: SystemLanguageModel
```

The base version of the model.

Instance Methods

```
func supportsLocale(Locale) -> Bool
```

Returns a Boolean indicating whether the given locale is supported by the model.

Relationships

Conforms To

Copyable
Observable
Sendable
SendableMetatype

See Also

Essentials

-  Generating content and performing tasks with Foundation Models

Enhance the experience in your app by prompting an on-device large language model.
-  Improving the safety of generative model output

Create generative experiences that appropriately handle sensitive inputs and respect people.
-  Support languages and locales with Foundation Models

Generate content in the language people prefer when they interact with your app.

{ } Adding intelligent app features with generative models

Build robust apps with guided generation and tool calling by adopting the Foundation Models framework.

```
struct UseCase
```

A type that represents the use case for prompting.