```
import Foundation

/// A compiled model asset.
///
/// `MLModelAsset` is an abstraction of a compiled model, which can be:
///
///  - `.mlmodelc` bundle on the file system
///  - In-memory model specification
///
/// It provides the unified interface to query the model description and to instantiate
`MLModel`.
///
/// ```swift
/// // Creates an object.
/// let modelAsset = MLModelAsset(url: modelURL)
///
/// // Query the model description
/// let description = try await modelAsset.modelDescription
///
/// // Query the list of functions in the model asset.
/// let functionNames = try await modelAsset.functionNames
///
/// // Query the model description of a specific function.
/// let descriptionOfMyFunction = try await
modelAsset.modelDescription(of: "MyFunction")
///
/// // Instantiate `MLModel` for "MyFunction".
/// let modelConfiguration = MLModelConfiguration()
/// modelConfiguration.functionName = "MyFunction"
/// let model = try await MLModel.load(asset: modelAsset,
configuration: modelConfiguration)
/// ```
@available macOS 13.0
open class MLModelAsset   NSObject

    /// Construct a model asset from the contents of specification data.
    ///
    /// - Parameters:
    ///    - specificationData: Contents of .mlmodel as a data blob.
    ///    - error: When the model asset creation fails error is populated with
the reason for failure.
    @available macOS 13.0
    public convenience init
Data  throws

    /// Construct a model asset from an ML Program specification by replacing
blob file references with corresponding in-memory blobs.
    ///
    /// An ML Program may use `BlobFileValue` syntax, which stores the
blob data in external files and refers them by URL.
```

```
    /// This factory method enables in-memory workflow for such models by using the specified in-memory blob data in place of the external files.
    ///
    /// The format of in-memory blobs must be the same as the external files. The dictionary must contain all the reference URLs used in the specification.
    ///
    /// - Parameters:
    ///   - specification: Contents of .mlmodel as a data blob.
    ///   - blobMapping: A dictionary with blob URL as the key and blob data as the value.
    ///   - error: When the model asset creation fails error is populated with the reason for failure.
    @available macOS 15.0
    public convenience init
Data                URL   Data   throws

    /// Constructs a ModelAsset from a compiled model URL.
    ///
    /// - Parameters:
    ///   - compiledModelURL: Location on the disk where the model asset is present.
    ///   - error: Errors if the model asset is not loadable.
    ///
    /// - Returns: a model asset or nil if there is an error.
    @available macOS 15.0
    public convenience init                          URL  throws

    /// The default model descripton.
    ///
    /// Use this method to get the description of the model such as the feature descriptions, the model author, and other metadata.
    ///
    /// For the multi-function model asset, this method vends the description for the default function. Use `modelDescription(for:)` to get the model description of other functions.
    ///
    /// ```swift
    /// let modelAsset = try MLModelAsset(url: modelURL)
    /// let modelDescription = try await modelAsset.modelDescription()
    /// print(modelDescription)
    /// ```
    @available macOS 15.0
    open func modelDescription
@escaping  MLModelDescription    any Error       Void

    /// The default model descripton.
    ///
    /// Use this method to get the description of the model such as the feature descriptions, the model author, and other metadata.
```

```
    ///
    /// For the multi-function model asset, this method vends the description for the default function. Use `modelDescription(for:)` to get the model description of other functions.
    ///
    /// ```swift
    /// let modelAsset = try MLModelAsset(url: modelURL)
    /// let modelDescription = try await modelAsset.modelDescription()
    /// print(modelDescription)
    /// ```
    @available macOS 15.0
    open var modelDescription  MLModelDescription   get async throws

    /// The model descripton for a specified function.
    ///
    /// Use this method to get the description of the model such as the feature descriptions, the model author, and other metadata.
    ///
    /// ```swift
    /// let modelAsset = try MLModelAsset(url: modelURL)
    /// let modelDescription = try await modelAsset.modelDescription(of: "my_function")
    /// print(modelDescription)
    /// ```
    @available macOS 15.0
    open func modelDescription
String                              @escaping
 MLModelDescription    any Error       Void

    /// The model descripton for a specified function.
    ///
    /// Use this method to get the description of the model such as the feature descriptions, the model author, and other metadata.
    ///
    /// ```swift
    /// let modelAsset = try MLModelAsset(url: modelURL)
    /// let modelDescription = try await modelAsset.modelDescription(of: "my_function")
    /// print(modelDescription)
    /// ```
    @available macOS 15.0
    open func modelDescription                    String  async throws    MLModelDescription

    /// The list of function names in the model asset.
    ///
    /// Some model types (e.g. ML Program) supports multiple functions. Use this method to query the function names.
```

```
    ///
    /// The method vends the empty array when the model doesn't use the multi-function configuration.
    ///
    /// ```swift
    /// let modelAsset = try MLModelAsset(url: modelURL)
    /// let functionNames = try await modelAsset.functionNames
    /// print(functionNames) // For example, ["my_function1", "my_function2"];
    /// ```
    @available macOS 15.0
    open func functionNames
@escaping   String     any Error       Void

    /// The list of function names in the model asset.
    ///
    /// Some model types (e.g. ML Program) supports multiple functions. Use this method to query the function names.
    ///
    /// The method vends the empty array when the model doesn't use the multi-function configuration.
    ///
    /// ```swift
    /// let modelAsset = try MLModelAsset(url: modelURL)
    /// let functionNames = try await modelAsset.functionNames
    /// print(functionNames) // For example, ["my_function1", "my_function2"];
    /// ```
    @available macOS 15.0
    open var functionNames   String    get async throws
```