

[Authentication Services](#) / `ASCredentialExportManager`

## Class

# ASCredentialExportManager

A class to manage exporting credentials.

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

```
class ASCredentialExportManager
```

## Overview

`ASCredentialExportManager` allows your app to exchange authentication credentials like passwords and passkeys with other apps. Participating apps such as password managers can receive your exported credentials by using the [ASCredentialImportManager](#) class.

## Configure your app

To participate in credential exchange, edit your credential provider extension's information property list and add the following key with a Boolean value of YES:

```
NSExtension > NSExtensionAttributes > ASCredentialProviderExtension  
Capabilities > SupportsCredentialExchange
```

Also, declare the versions of the credential data format your app supports using the following key:

```
NSExtension > NSExtensionAttributes > ASCredentialProviderExtension  
Capabilities > SupportedCredentialExchangeVersions
```

The value is an array of strings containing every version your app supports. Currently the only available version is `1.0`.

## Export credentials

To export credentials, your app's interface needs to allow the person using it to select the credentials they want to export. To begin the process, call the `requestExport(for:)` method on an instance of this class. Calling this method brings up an out-of-process system UI that guides the person through the export procedure. The system UI explains the risks of exporting credentials, and then allows them to choose an app to export to. The operating system acts as an intermediary to establish the identities of the password manager apps involved and performs the exchange; this process doesn't write any data to the file system.

When a person chooses an importer app, the system launches it, sending an `NSUserActivity` whose `activityType` is `ASCredentialExchangeActivity`. The importing app responds to the launch activity by calling the `ASCredentialImportManager` method `importCredentials(token:)` to receive the exported credentials.

When a person chooses an importer app, the `requestExport(for:)` method returns an `ASCredentialExportManager.ExportOptions` structure that describes the credential data format version you should use to ensure the importer app can successfully decode the data. Use this version to construct your `AExportedCredentialData` object, then call `exportCredentials(_ :)` with it. After the system receives your data, the system launches the importer app, sending the `NSUserActivity` `ASCredentialExchangeActivity`. The importing app responds to the launch activity by calling the `ASCredentialImportManager` method `importCredentials(token:)` to receive the exported credentials.

The following example shows how to perform an export from a SwiftUI view. The example assumes the typical case where the app has a single credential provider extension, which means it can pass `nil` to `requestExport(for:)`.

```
struct CredentialExportManagerExample: View {
    @Environment(\.credentialExportManager) private var credentialExportManager

    var body: some View {
        Button("Export Credentials") {
            Task {
                do {
                    let exportOptions = try await credentialExportManager.requestExportOptions()
                    let credentialData = getCredentialData(exportOptions: exportOptions)
                    try await credentialExportManager.exportCredentials(credentialData)
                } catch {
                    // Handle the export error.
                }
            }
        }
    }
}
```

For a corresponding import code example, see [ASCredentialImportManager](#).

---

## Topics

### Creating an export manager

`convenience init(presentationAnchor: ASPresentationAnchor)`

Creates an export manager, anchored by the given AppKit window.

`convenience init(presentationAnchor: ASPresentationAnchor)`

Creates an export manager, anchored by the given UIKit window.

`typealias ASPresentationAnchor`

A platform-specific type that indicates the kind of user interface element to use as a presentation anchor.

### Exporting credentials

`func requestExport(for: String?) async throws -> ASCredentialExportManager.ExportOptions`

Begins the export process.

`struct ExportOptions`

Options that configure the behavior of a credential export operation.

`func exportCredentials(ASExportedCredentialData) async throws`

Exports the provided credential data.

`struct ASExportedCredentialData`

A container for credential data that your app provides to an exporter or receives from an importer.

---

## See Also

### Credential migration

`class ASCredentialImportManager`

A class to manage importing credentials.