

[Authentication Services](#) / `ASCredentialImportManager`

## Class

# ASCredentialImportManager

A class to manage importing credentials.

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

```
class ASCredentialImportManager
```

## Overview

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

## Configuring 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`.

## Importing credentials

Credential export begins when another app calls `exportCredentials(_:)`, which brings up a system UI to choose an app to export to. When the person using the export app chooses your app to receive the credentials, the system launches your app and sends an `NSUserActivity` whose `activityType` is `ASCredentialExchangeActivity`. To support this process, add the `NSUserActivityTypes` array to your app's information property list and add the item `ASCredentialExchangeActivityType` to the array.

Your app needs to prepare for the system launching the app from this activity by fetching the `ASCredentialImportToken` from the activity's `userInfo` dictionary. The value of this key is a `UUID` token; create an instance of `ASCredentialImportManager` and pass the token to `importCredentials(token:)` to begin the import process.

The following example shows how a SwiftUI app handles launching from the user activity and beginning the credential import process.

```
struct CredentialImportManagerExample: View {
    @Environment(\.credentialImportManager) private var credentialImportManager

    var body: some View {
        content // Defined elsewhere.
        .onContinueUserActivity(ASCredentialExchangeActivityType) { activity in
            Task {
                do {
                    guard let token = activity.userInfo?[ASCredentialImportTokenKey] as? UUID
                    let credentialData = try await credentialImportManager.importCredentials(token: token)
                    // Do something with the data.
                } catch {
                    // Handle the import error.
                }
            }
        }
    }
}
```

For a corresponding export code example, see [ASCredentialExportManager](#).

---

## Topics

### Creating an import manager

`init()`

Creates an import manager instance.

## Importing credentials

```
func importCredentials(token: UUID) async throws -> AExported  
CredentialData
```

Begins the credential import process.

```
struct AExportedCredentialData
```

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

---

## See Also

### Credential migration

```
class ASCredentialExportManager
```

A class to manage exporting credentials.