

[Foundation](#) / [UserDefaults](#) / `init(suiteName:)`

Initializer

`init(suiteName:)`

Creates a user defaults object initialized with the defaults for the specified database name.

iOS 7.0+ | iPadOS 7.0+ | Mac Catalyst 13.1+ | macOS 10.9+ | tvOS 9.0+ | visionOS 1.0+ | watchOS 2.0+

```
init?(suiteName suiteName: String?)
```

Parameters

`suiteName`

The domain identifier of the search list.

If you pass `nil` to this parameter, the system uses the default search list that the [standard](#) class method uses. Because a suite manages the defaults of a specified app group, a suite name must be distinct from your app's main bundle identifier. The [globalDomain](#) is also an invalid suite name, because it isn't writeable by apps.

Discussion

You can use this method when developing an app suite, to share preferences or other data among the apps, or when developing an app extension, to share preferences or other data between the extension and its containing app.

The argument and registration domains are shared between all instances of [UserDefaults](#).

The `suiteName` parameter matches the domain parameter of the corresponding CFPreferences APIs (except when translating between Foundation and Core Foundation constants). The code example below shows two equivalent statements. For more details, see [Preferences Utilities](#).

Equivalent statements using `NSUserDefaults` and CFPreferences APIs

```
let userDefaultsValue = UserDefaults(suiteName: "someDomain")?.object(forKey: "someKey")
let preferencesValue = CFPREFERENCESCopyAppValue("someKey" as CFString, "someDomain")
// userDefaultsValue and preferencesValue are equal
```

On macOS, specifying another app's bundle identifier will get you that app's preferences search list, unless prevented by the [App Sandbox](#).

See Also

Creating User Defaults Objects

`convenience init()`

Creates a user defaults object initialized with the defaults for the app and current user.