Documentation

Language: **Swift**

Contacts / CNContact

Class

# CNContact

An immutable object that stores information about a single contact, such as the contact's first name, phone numbers, and addresses.

iOS 9.0+ | iPadOS 9.0+ | Mac Catalyst 13.1+ | macOS 10.11+ | visionOS 1.0+ | watchOS 2.0+

```
class CNContact
```

## Overview

A `CNContact` object stores an immutable copy of a contact's information, so you cannot change the information in this object directly. Contact objects are thread-safe, so you may access them from any thread of your app.

To modify a contact's information, call the `mutableCopy()` method to obtain a `CNMutableContact` object with the same information. After modifying the mutable contact, save your changes back to the contacts database using the `CNContactStore` object.

Every contact in the contacts database has a unique ID, which you access using the `identifier` property. The mutable and immutable versions of the same contact have the same identifier.

## Topics

### Identifying the Contact

```
var identifier: String
```

A value that uniquely identifies a contact on the device.

```
var contactType: CNContactType
```

An enum identifying the contact type.

```
enum CNContactType
```

The types a contact can be.

## Getting Name Information

```
var namePrefix: String
```

The name prefix of the contact.

```
var givenName: String
```

The given name of the contact.

```
var middleName: String
```

The middle name of the contact.

```
var familyName: String
```

The family name of the contact.

```
var previousFamilyName: String
```

A string for the previous family name of the contact.

```
var nameSuffix: String
```

The name suffix of the contact.

```
var nickname: String
```

The nickname of the contact.

```
var phoneticGivenName: String
```

The phonetic given name of the contact.

```
var phoneticMiddleName: String
```

The phonetic middle name of the contact.

```
var phoneticFamilyName: String
```

A string for the phonetic family name of the contact.

## Getting Work Information

```
var jobTitle: String
```

The contact’s job title.

```
var departmentName: String
```

The name of the department associated with the contact.

```
var organizationName: String
```

The name of the organization associated with the contact.

```
var phoneticOrganizationName: String
```

The phonetic name of the organization associated with the contact.

## Getting Addresses

```
var postalAddresses: [CNLabeledValue<CNPostalAddress>]
```

An array of labeled postal addresses for a contact.

```
var emailAddresses: [CNLabeledValue<NSString>]
```

An array of labeled email addresses for the contact.

```
var urlAddresses: [CNLabeledValue<NSString>]
```

An array of labeled URL addresses for a contact.

## Getting Phone Information

```
var phoneNumbers: [CNLabeledValue<CNPhoneNumber>]
```

An array of labeled phone numbers for a contact.

## Getting Social Profiles

```
var socialProfiles: [CNLabeledValue<CNSocialProfile>]
```

An array of labeled social profiles for a contact.

## Getting Birthday Information

```
var birthday: DateComponents?
```

A date component for the Gregorian birthday of the contact.

```
var nonGregorianBirthday: DateComponents?
```

A date component for the non-Gregorian birthday of the contact.

```
var dates: [CNLabeledValue<NSDateComponents>]
```

An array containing labeled Gregorian dates.

## Getting Notes

```
var note: String
```

A string containing notes for the contact.

## Getting Contact Images

```
var imageData: Data?
```

The profile picture of a contact.

```
var thumbnailImageData: Data?
```

The thumbnail version of the contact's profile picture.

```
var imageDataAvailable: Bool
```

A Boolean indicating whether a contact has a profile picture.

## Getting Related Information

```
var contactRelations: [CNLabeledValue<CNContactRelation>]
```

An array of labeled relations for the contact.

```
var instantMessageAddresses: [CNLabeledValue<CNInstantMessageAddress>]
```

An array of labeled IM addresses for the contact.

## Localizing Contact Data

```
class func localizedString(forKey: String) -> String
```

Returns a string containing the localized contact property name.

## Comparing Contacts

```
class func descriptorForAllComparatorKeys() -> any CNKeyDescriptor
```

Fetches all the keys required for the contact sort comparator.

```
class func comparator(forNameSortOrder: CNContactSortOrder) ->
Comparator
```

Returns a comparator to sort contacts with the specified order.

```
func isUnifiedWithContact(withIdentifier: String) -> Bool
```

Returns a Boolean indicating whether the current contact is a unified contact and includes a contact with the specified identifier.

```
enum CNContactSortOrder
```

Indicates the sorting order for contacts.

## Checking the Availability of Data

```
func isKeyAvailable(String) -> Bool
```

Determines whether the contact property value for the specified key is fetched.

```
func areKeysAvailable([any CNKeyDescriptor]) -> Bool
```

Determines whether all contact property values for the specified keys are fetched.

## Getting Search Predicates

Predicates to match contacts. You can only use these predicates with `CNContactStore` and `CNContactFetchRequest`.

```
class func predicateForContacts(matchingName: String) -> NSPredicate
```

Returns a predicate to find the contacts matching the specified name.

```
class func predicateForContacts(withIdentifiers: [String]) -> NSPredicate
```

Returns a predicate to find the contacts matching the specified identifiers.

```
class func predicateForContactsInGroup(withIdentifier: String) -> NSPredicate
```

Returns a predicate to find the contacts that are members in the specified group.

```
class func predicateForContactsInContainer(withIdentifier: String) -> NSPredicate
```

Returns a predicate to find the contacts in the specified container.

```
class func predicateForContacts(matching: CNPhoneNumber) -> NSPredicate
```

Returns a predicate to find the contacts whose phone number matches the specified value.

```
class func predicateForContacts(matchingEmailAddress: String) -> NSPredicate
```

Returns a predicate to find the contacts whose email address matches the specified value.

## Relationships

### Inherits From

`NSObject`

### Inherited By

`CNMutableContact`

### Conforms To

`CVarArg`
`Copyable`
`CustomDebugStringConvertible`
`CustomStringConvertible`
`Equatable`
`Hashable`
`Identifiable`
`NSCoding`
`NSCopying`
`NSItemProviderReading`
`NSItemProviderWriting`
`NSMutableCopying`
`NSObjectProtocol`
`NSSecureCoding`

## See Also

### Contact data

`class CNMutableContact`

A mutable object that stores information about a single contact, such as the contact’s first name, phone numbers, and addresses.

Data Objects

Access contact-related data, such as the user’s postal address and phone number.

Contact Keys

Specify contact-related properties during fetch operations.