
iChat Instant Message Framework Reference



2006-05-23



Apple Computer, Inc.
© 2004, 2006 Apple Computer, Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Computer, Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Computer, Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled or Apple-licensed computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Computer, Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Apple, the Apple logo, Bonjour, Cocoa, Mac, and Mac OS are trademarks of Apple Computer, Inc., registered in the United States and other countries.

Objective-C is a registered trademark of NeXT Software, Inc.

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR

IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

Introduction [The InstantMessage Framework](#) 5

Part I [Classes](#) 7

Chapter 1 [IMService Class Reference](#) 9

[Class Description](#) 9
[Methods by Task](#) 9
[Class Methods](#) 11
[Instance Methods](#) 12
[Constants](#) 14
[Notifications](#) 16

Part II [Data Types](#) 19

Chapter 2 [Instant Message Data Types Reference](#) 21

[Introduction](#) 21
[Data Types](#) 21

Part III [Constants](#) 23

Chapter 3 [Instant Message Constants Reference](#) 25

[Introduction](#) 25
[Constants](#) 25

[Document Revision History](#) 29

[Index](#) 31

C O N T E N T S

The InstantMessage Framework

| | |
|--------------------------------|---|
| Framework | /System/Library/Frameworks/InstantMessage.framework |
| Header file directories | /System/Library/Frameworks/InstantMessage.framework/Headers |

The IMService API provides a way to integrate a variety of data about a user's iChat connections into your application. It provides information on which services the user is connected to (AIM or Bonjour), their online screen names, their lists of acquaintances, their current status on a given service (away, idle, available), idle times, and other presence-specific details. The API also provides notifications to update your applications when a user's status, information, status images, or service connections have changed. A variety of status notifications related to the user's status and preferences are posted by the IMService custom notification center. See the "Notifications" section in [IMService](#) (page 9) for more information.

The InstantMessage framework can be used only by an Objective-C Cocoa application.

I N T R O D U C T I O N

The InstantMessage Framework

Classes

IMService Class Reference

| | |
|-----------------------|---|
| Inherits from: | NSObject |
| Conforms to: | NSObject (NSObject) |
| Framework | /System/Library/Frameworks/InstantMessage.framework |
| Declared in: | InstantMessage/IMService.h |

Class Description

Each IMService object represents one service available through iChat. Class methods such as `allServices` and `serviceName:` will return these objects. Each object acts as the liaison to a single instant messaging service, allowing you to access the individual user's global status, the user's list of acquaintances, and other information which can be integrated into your application. A variety of status notifications related to the user's status and preferences are posted by the IMService custom notification center.

Methods by Task

Accessing instant messaging services

- + `allServices` (page 11)
Returns an NSArray of IMService objects corresponding to the current available services (AIM, Bonjour, and so on.).
- + `serviceName:` (page 12)
Returns the IMService object corresponding to the service specified by *name*.

Accessing service attributes

- + [imageURLForStatus:](#) (page 11)
Returns the URL of the image corresponding to the `IMPersonStatus` specified by *status*.
- + [myIdleTime](#) (page 11)
Returns a `NSDate`, in seconds, of time since the currently active user went idle.
- + [myStatus](#) (page 11)
Returns an `IMPersonStatus` object representing the status of the currently active user.
- + [notificationCenter](#) (page 12)
Returns the custom notification center for the Instant Messaging service.
- [localizedName](#) (page 13)
Returns an `NSString` containing the user-visible localized name of the service, such as "AOL Instant Messenger" or "Bonjour", for example.
- [localizedShortName](#) (page 13)
- [name](#) (page 13)
Returns an `NSString` containing the fixed canonical name of the service.
- [status](#) (page 14)
Returns the appropriate `IMServiceStatus` number for the current login status of the service.

Accessing people

- [peopleWithScreenName:](#) (page 14)
Returns an array of Address Book `ABPerson` objects that correspond to the person with the screen name matched by *screenName*.
- [screenNamesForPerson:](#) (page 14)
Returns an array of strings that are valid screen names for the person specified by *person*.

Accessing person attributes

- [infoForAllScreenNames](#) (page 12)
Returns an array of information about all people and all accounts currently logged in to the service.
- [infoForPreferredScreenNames](#) (page 12)
Returns an array of information about all people and their primary accounts currently logged in to the service.
- [infoForScreenName:](#) (page 13)
Returns an `NSDictionary` with information content about the account specified by *screenName*.

Class Methods

allServices

Returns an NSArray of IMService objects corresponding to the current available services (AIM, Bonjour, and so on.).

```
+ (NSArray *)allServices
```

Availability

Available in Mac OS X version 10.4 and later.

imageURLForStatus:

Returns the URL of the image corresponding to the IMPersonStatus specified by *status*.

```
+ (NSURL *)imageURLForStatus:(IMPersonStatus)status
```

Discussion

This image reflects the current online status of the user, and is usually reflected by a colored bubble or triangle. See the IMPersonStatus information in the “Types and Constants” section of this reference.

Availability

Available in Mac OS X version 10.4 and later.

myIdleTime

Returns a NSDate, in seconds, of time since the currently active user went idle.

```
+ (NSDate *)myIdleTime
```

Availability

Available in Mac OS X version 10.4 and later.

myStatus

Returns an IMPersonStatus object representing the status of the currently active user.

```
+ (IMPersonStatus)myStatus
```

Discussion

This status is global across all services. See the IMPersonStatus information in the “Types and Constants” section of this reference.

Availability

Available in Mac OS X version 10.4 and later.

notificationCenter

Returns the custom notification center for the Instant Messaging service.

```
+ (NSNotificationCenter *)notificationCenter
```

Discussion

This custom notification center manages the notifications specified in [“Notifications”](#) (page 16).

Availability

Available in Mac OS X version 10.4 and later.

serviceWithName:

Returns the IMService object corresponding to the service specified by *name*.

```
+ (IMService *)serviceWithName:(NSString *)name;
```

Discussion

The parameter *name* is an NSString containing a service name as returned by a previous call to [name](#) (page 13). Hard-coding the service names is not recommended.

Availability

Available in Mac OS X version 10.4 and later.

Instance Methods

infoForAllScreenNames

Returns an array of information about all people and all accounts currently logged in to the service.

```
- (NSArray *)infoForAllScreenNames
```

Discussion

If a single person is logged in on multiple accounts (determined by the user's Address Book), this method will return the information for all of the logged-in accounts. The return type is an NSArray of the dictionaries returned by [infoForScreenName:](#) (page 13) for all people on all accounts.

Availability

Available in Mac OS X version 10.4 and later.

infoForPreferredScreenNames

Returns an array of information about all people and their primary accounts currently logged in to the service.

```
- (NSArray *)infoForPreferredScreenNames
```

Discussion

If a single person is logged in on multiple accounts (determined by the user's Address Book), this method will only return the information for the preferred account. The preferred account is determined by iChat, using a combination of capabilities (video chat capability, audio chat capability, and so on), status (available, idle, away), and other user attributes.

Availability

Available in Mac OS X version 10.4 and later.

infoForScreenName:

Returns an NSDictionary with information content about the account specified by *screenName*.

```
- (NSDictionary *)infoForScreenName: (NSString *)screenName
```

Discussion

The information in the dictionary is specified by the dictionary keys specified in [“Constants”](#) (page 14).

Availability

Available in Mac OS X version 10.4 and later.

localizedName

Returns an NSString containing the user-visible localized name of the service, such as "AOL Instant Messenger" or "Bonjour", for example.

```
- (NSString *)localizedName
```

Discussion

This string will be localized if required.

Availability

Available in Mac OS X version 10.4 and later.

localizedShortName

```
- (NSString *)localizedShortName
```

Discussion

Returns a shorter version, if available, of the user-visible localized name of the service. This string will be localized if required.

Availability

Available in Mac OS X version 10.4 and later.

name

Returns an NSString containing the fixed canonical name of the service.

```
- (NSString *)name;
```

Discussion

This string is not localized.

Availability

Available in Mac OS X version 10.4 and later.

peopleWithScreenName:

Returns an array of Address Book ABPerson objects that correspond to the person with the screen name matched by *screenName*.

```
- (NSArray *)peopleWithScreenName:(NSString *)screenName
```

Discussion

Can return an empty array or an array with one or more items.

Availability

Available in Mac OS X version 10.4 and later.

screenNamesForPerson:

Returns an array of strings that are valid screen names for the person specified by *person*.

```
- (NSArray *)screenNamesForPerson:(ABPerson *)person
```

Discussion

See Address Book documentation for more information on ABPerson and accessing the user's address book. Can return an empty array or an array with one or more items.

Availability

Available in Mac OS X version 10.4 and later.

status

Returns the appropriate IMServiceStatus number for the current login status of the service.

```
- (IMServiceStatus)status
```

Discussion

See IMServiceStatus documentation for more information.

Availability

Available in Mac OS X version 10.4 and later.

Constants

These keys describe a person (accessed by [infoForScreenName:](#) (page 13)):

| Constant | Description |
|---------------------------|--|
| IMPersonAVBusyKey | NSNumber set to 0 if the person's audio/video capabilities are available, or 1 if they are busy. |
| IMPersonCapabilitiesKey | NSArray of the person's capabilities. |
| IMPersonEmailKey | NSString containing the person's e-mail address. This is a key used directly by Bonjour; however, if a person has an Address Book entry associated with a relevant AIM account, this key will reflect the first e-mail address of that person. |
| IMPersonFirstNameKey | NSString containing the person's first name. This is a key used directly by Bonjour; however, if a person has an Address Book entry associated with a relevant AIM account, this key will reflect the first name of that person. |
| IMPersonIdleSinceKey | NSDate containing the time, in seconds, since the last user activity. Available if the user's status is idle. |
| IMPersonLastNameKey | NSString containing the person's last name. This is a key used directly by Bonjour; however, if a person has an Address Book entry associated with a relevant AIM account, this key will reflect the last name of that person. |
| IMPersonPictureDataKey | NSData containing the image that should be used for the user's icon. |
| IMPersonScreenNameKey | NSString containing the service-specific identifier for a person. Examples include "User123" or "steve@mac.com" for AIM, or "John Doe" for Bonjour. |
| IMPersonServiceNameKey | NSString containing the name of the service this person belongs to. |
| IMPersonStatusKey | NSNumber containing the IMPersonStatus representing the current online state of the person, if known. See the IMPersonStatus documentation for more information. |
| IMPersonStatus-MessageKey | NSString containing the user's current status message. |

These keys describe a person's client capabilities, accessed with the [IMPersonCapabilitiesKey](#) (page 15):

| Constant | Description |
|-----------------------------|---------------------------|
| IMCapabilityAudioConference | Audio chat capability |
| IMCapabilityDirectIM | Direct connect capability |
| IMCapabilityFileSharing | File sharing capability |
| IMCapabilityFileTransfer | File transfer capability |
| IMCapabilityText | Text capability |

| Constant | Description |
|-----------------------------|-----------------------|
| IMCapabilityVideoConference | Video chat capability |

These keys describe service status states:

| Constant | Description |
|-----------------------------|--|
| IMServiceStatusLoggedOut | Service is currently logged out |
| IMServiceStatusDisconnected | Service was disconnected, not by user but by system or error condition |
| IMServiceStatusLoggingOut | Service is in the process of logging out |
| IMServiceStatusLoggingIn | Service is in the process of logging in |
| IMServiceStatusLoggedIn | Service is currently logged in |

These keys describe person status states (global across all services):

| Constant | Description |
|-------------------------|--------------------------------------|
| IMPersonStatusUnknown | Person's status is currently unknown |
| IMPersonStatusOffline | Person is currently offline |
| IMPersonStatusIdle | Person is currently idle |
| IMPersonStatusAway | Person is currently away |
| IMPersonStatusAvailable | Person is currently available |

Notifications

IMPersonInfoChangedNotification

Received from IMService's custom notification center. Posted when a screen name changes some aspect of their published information. The user information dictionary will always contain an `IMPersonScreenNameKey` and may contain any of the following keys as described by “Constants” (page 14): `IMPersonStatusMessageKey`, `IMPersonIdleSinceKey`, `IMPersonFirstNameKey`, `IMPersonLastNameKey`, `IMPersonEmailKey`, `IMPersonPictureDataKey`, `IMPersonAVBusyKey`, or `IMPersonCapabilitiesKey`. If a particular attribute has been removed, the value for the relevant key will be `NSNull`.

IMPersonStatusChangedNotification

Received from IMService's custom notification center. Posted when a different user (screen name) logs in, logs off, goes away, and so on. This notification is for the IMService object. The user information

dictionary will always contain an `IMPersonScreenNameKey` and an `IMPersonStatusKey`, and no others.

IMServiceStatusChangedNotification

Received from IMService's custom notification center. Posted when the user logs in, logs off, goes away, and so on. This notification is relevant to no particular object. The user information dictionary will not contain keys. The client should call `status` (page 14) to get the new service status.

IMStatusImagesChangedAppearanceNotification

Received from IMService's custom notification center. Posted when the user changes their preferred images for displaying status. This notification is relevant to no particular object. The user information dictionary will not contain keys. Clients that display status information graphically (using the green/yellow/red dots and triangles) should call `imageURLForStatus:` (page 11) to get the new image.

Data Types

Instant Message Data Types Reference

Framework: InstantMessage/IMService.h

Introduction

This document describes the data types found in the Instant Message framework.

Data Types

IMPersonStatus

Online status states for a person (accessed using the `IMPersonStatus` key).

```
typedef enum {
    IMPersonStatusUnknown,
    IMPersonStatusOffline,
    IMPersonStatusIdle,
    IMPersonStatusAway,
    IMPersonStatusAvailable
} IMPersonStatus;
```

Discussion

These states are described in [“Constants”](#) (page 14) in `IMService`.

Availability

Available in Mac OS X version 10.4 and later.

IMServiceStatus

Status states for the service (accessed using the `status` instance method).

```
typedef enum {
    IMServiceStatusLoggedOut,
    IMServiceStatusDisconnected,
    IMServiceStatusLoggingOut,
}
```

```
    IMServiceStatusLoggingIn,  
    IMServiceStatusLoggedIn  
} IMServiceStatus;
```

Discussion

These states are described in [“Constants”](#) (page 14) in `IMService`.

Availability

Available in Mac OS X version 10.4 and later.

Constants

Instant Message Constants Reference

Framework: InstantMessage/IMService.h

Introduction

This document describes the constants found in the Instant Message framework.

Constants

Properties

Client Properties

Capabilities that the relevant screen name's client supports.

```
extern NSString *IMCapabilityText;  
extern NSString *IMCapabilityDirectIM;  
extern NSString *IMCapabilityFileTransfer;  
extern NSString *IMCapabilityFileSharing;  
extern NSString *IMCapabilityAudioConference;  
extern NSString *IMCapabilityVideoConference;
```

Discussion

These properties are described in [“Constants”](#) (page 14) in IMService.

Availability

Available in Mac OS X version 10.4 and later.

Screen Name Properties

Keys describing a particular screen name.

```
extern NSString *IMPersonServiceNameKey;
```

Instant Message Constants Reference

```
extern NSString *IMPersonScreenNameKey;
extern NSString *IMPersonStatusKey;
extern NSString *IMPersonStatusMessageKey;
extern NSString *IMPersonIdleSinceKey;
extern NSString *IMPersonFirstNameKey;
extern NSString *IMPersonLastNameKey;
extern NSString *IMPersonEmailKey;
extern NSString *IMPersonPictureDataKey;
extern NSString *IMPersonAVBusyKey;
extern NSString *IMPersonCapabilitiesKey;
```

Discussion

These properties are described in [“Constants”](#) (page 14) in `IMService`.

Availability

Available in Mac OS X version 10.4 and later.

Notifications

IMPersonInfoChangedNotification

Posted when a screen name changes some aspect of their published information.

```
extern NSString *IMPersonInfoChangedNotification;
```

Discussion

This notification is described in [“Notifications”](#) (page 16) in `IMService`.

Availability

Available in Mac OS X version 10.4 and later.

IMPersonStatusChangedNotification

Posted when a different user (screen name) logs in, logs off, goes away, and so on.

```
extern NSString *IMPersonStatusChangedNotification;
```

Discussion

This notification is described in [“Notifications”](#) (page 16) in `IMService`.

Availability

Available in Mac OS X version 10.4 and later.

IMServiceStatusChangedNotification

Posted when the user logs in, logs off, goes away, and so on.

```
extern NSString *IMServiceStatusChangedNotification;
```

Discussion

This notification is described in [“Notifications”](#) (page 16) in `IMService`.

Availability

Available in Mac OS X version 10.4 and later.

IMStatusImagesChangedAppearanceNotification

Posted when the user changes their preferred images for displaying status.

```
extern NSString *IMStatusImagesChangedAppearanceNotification;
```

Discussion

This notification is described in [“Notifications”](#) (page 16) in *IMService*.

Availability

Available in Mac OS X version 10.4 and later.

Document Revision History

This table describes the changes to *iChat Instant Message Framework Reference*.

| Date | Notes |
|------------|--|
| 2006-05-23 | First publication of this content as a collection of separate documents. |

REVISION HISTORY

Document Revision History

Index

A

`allServices` class method [11](#)

C

`Client Properties` [25](#)

I

`imageURLForStatus:` class method [11](#)
`IMCapabilityAudioConference` constant [15](#)
`IMCapabilityDirectIM` constant [15](#)
`IMCapabilityFileSharing` constant [15](#)
`IMCapabilityFileTransfer` constant [15](#)
`IMCapabilityText` constant [15](#)
`IMCapabilityVideoConference` constant [16](#)
`IMPersonAVBusyKey` constant [15](#)
`IMPersonCapabilitiesKey` constant [15](#)
`IMPersonEmailKey` constant [15](#)
`IMPersonFirstNameKey` constant [15](#)
`IMPersonIdleSinceKey` constant [15](#)
`IMPersonInfoChangedNotification` [26](#)
`IMPersonInfoChangedNotification` notification [16](#)
`IMPersonLastNameKey` constant [15](#)
`IMPersonPictureDataKey` constant [15](#)
`IMPersonScreenNameKey` constant [15](#)
`IMPersonServiceNameKey` constant [15](#)
`IMPersonStatus` data type [21](#)
`IMPersonStatusAvailable` constant [16](#)
`IMPersonStatusAway` constant [16](#)
`IMPersonStatusChangedNotification` [26](#)
`IMPersonStatusChangedNotification` notification [16](#)
`IMPersonStatusIdle` constant [16](#)
`IMPersonStatusKey` constant [15](#)
`IMPersonStatusMessageKey` constant [15](#)
`IMPersonStatusOffline` constant [16](#)

`IMPersonStatusUnknown` constant [16](#)
`IMServiceStatus` data type [21](#)
`IMServiceStatusChangedNotification` [26](#)
`IMServiceStatusChangedNotification` notification [17](#)
`IMServiceStatusDisconnected` constant [16](#)
`IMServiceStatusLoggedIn` constant [16](#)
`IMServiceStatusLoggedOut` constant [16](#)
`IMServiceStatusLoggingIn` constant [16](#)
`IMServiceStatusLoggingOut` constant [16](#)
`IMStatusImagesChangedAppearanceNotification` [27](#)
`IMStatusImagesChangedAppearanceNotification` notification [17](#)
`infoForAllScreenNames` instance method [12](#)
`infoForPreferredScreenNames` instance method [12](#)
`infoForScreenName:` instance method [13](#)

L

`localizedName` instance method [13](#)
`localizedShortName` instance method [13](#)

M

`myIdleTime` class method [11](#)
`myStatus` class method [11](#)

N

`name` instance method [13](#)
`notificationCenter` class method [12](#)

P

`peopleWithScreenName:` instance method [14](#)

S

Screen Name Properties [25](#)
screenNamesForPerson: [instance method 14](#)
serviceWithName: [class method 12](#)
status [instance method 14](#)