

Getting Started with Game Center

Overview

iOS 4.1 includes Game Center, Apple's new social gaming network. Game Center enables your users to track their best scores on a leaderboard, compare their achievements, invite friends to play a game, and start a multiplayer game through auto-matching.

The functionality of Game Center is delivered in three parts:

- The Game Center application is the central starting point where users create their own account. Users sign in to Game Center and interact with other players anonymously using their nickname. Players can set status messages, invite others to be their friends, see which games their friends have played recently, invite friends to play a game, start a multiplayer game through auto-matching, compare their achievements, track their best scores, and much more.
- The Game Kit APIs allow game developers to create apps that will work with Game Center. Your games can authenticate the player and get information about their friends, report and display scores, unlock achievements, provide multi-player functionality, and even manage voice chat between players.
- The Game Center service is an online service provided by Apple to perform player authentication, provide leaderboard and achievement data, carry out auto-matching for multi-player games, and many other services.

This document describes in detail the approach for supporting Game Center in your app and gives you the information you need to create a social gaming experience that users find compelling, fun, and easy to use.

Using the Game Kit API

Authenticating the player

Game Center allows each user (or player) to create an account that is used for authentication. Before using other Game Center features, your application must first authenticate the user.

Typically, after the user has entered their Game Center authentication credentials once, these remain in effect *indefinitely* for *all* Game Center enabled apps on that device. All apps can subsequently authenticate the user without requiring their credentials to be re-entered. Authentication on that device ends if the user elects to log out from the Game Center app.

Your app should authenticate the player as soon as UI can be presented so that you can readily present Game Center information such as leaderboards and achievements, or initiate multi-player games as soon as the user requests it.

It is important to check for errors when authenticating. The user may choose not to authenticate or they may be using a device that does not support Game Center. Your app must be prepared to fall-back gracefully in each case and the game should proceed normally with Game Center features disabled, if it is able to do so.

After authenticating, Game Center will provide your app with the user's unique player identifier, nickname, and other information. The player identifier is used throughout the Game Kit API whenever data specific to that player needs to be stored by Game Center and accessed by your application, such as to post a high score or report an achievement. Your application may also use the player identifier to associate the user with other custom information that you store in the game. However, you should never present or reveal the player identifier to the player or to a third party.

It's important to remember that more than one person might play Game Center enabled games on the same device. If your app stores game data you should consider indexing this data on a per-user basis, associated with each user's unique player identifier.

Also note that the current user may suspend your app (or move it to the background) at any point, sign in as another player from the Game Center app, and then resume your game. Your app will receive a notification of a player change in such an event and you must implement a notification handler to allow your app to respond appropriately.

Using Leaderboards

Leaderboards are a great way to create a sense of competition between players in your game and keep them wanting to come back to play more. Players can compare their scores with their friends or other players in Game Center, and see their ranking versus other scores reported that day, over the past week, or all-time.

Game Center provides your application with one overall leaderboard and up to 25 additional leaderboards distinguished by "category". Categories allow each of your game modes to have its own list of best scores. For example, different categories might be specified for game difficulty (easy, medium, hard), for each level or track, and so on.

If you specify leaderboards with more than one category, the category marked as the *default* category is the leaderboard shown for your game in the Game Center application. You may also choose to display an aggregate leaderboard that includes scores reported to *any* category in your application.

Always check for network errors when posting scores to Game Center. If there was an error, it is important for your app to store any leaderboard updates for later retransmission. This retransmission should occur at the soonest opportunity and without requiring separate user action.

Your app may present its leaderboard to the user using the standard Game Center leaderboard interface, or for maximum integration into the look and feel of your game you may elect to retrieve leaderboard data programmatically from Game Center and present it using a custom interface you develop.

To maintain an optimal user experience, your app should only retrieve the portion of the leaderboard data it expects to display, such as the Top 25 scores, the scores of players in the current match, or the scores ranked nearby the player's score. Do not attempt to retrieve the entire leaderboard for your game all at once.

Be sure to plan your allocation of leaderboards to take into account future game versions, additions made via In-App Purchases, and so on. Once a leaderboard has gone live for any version of your app, it can not be removed.

To request player scores be removed due to abuse or cheating, please contact [<itunesconnect@apple.com>](mailto:itunesconnect@apple.com).

Using Achievements

Achievements allow your application to create goals for your users, earned by reaching a milestone or performing an action. Achievements are another great way to increase competition between players in your game and keep them coming back to play over and over again. Achievements can give newer players a sense of accomplishment, and can provide veteran users of your game ideas for alternate approaches or advanced goals to attain.

For example, a first-person shooter might create achievements such as:

- Defeated the first-level boss
- Found the secret passageway
- Finished the first three levels without taking any damage
- Defeated all enemies in less than 3 minutes
- Played 10 team matches without dropping off

And so on. Some achievements may be very easy to get, and others could be very difficult and take days or weeks of practice.

Game Center allows each game to specify a maximum of 100 achievements.

Reporting Achievements

As the player makes progress towards completing an achievement your application should report their progress to Game Center as a percentage from 0% to 100%. When the progress reaches 100% the achievement is earned and will be displayed as completed when the user views their achievements.

It is up to you to decide how progress towards an achievement is calculated. For example, if your game requires the player to find ten objects, you might increment the percentage by 10% every time the user finds one. Alternatively, if the player earns an achievement for performing a specific action, then the one and only time you report the achievement might be when the player completes that action.

In addition to reporting achievement progress to Game Center, it is recommended that you also inform the user when they've earned an achievement. It is up to you to do so in a way that fits the style of your game. For example, you might present a dialog informing the user about an

achievement at the moment progress reaches 100%, or wait for a break in the action and present a list of achievements recently earned, or by another means of your choosing.

Always check for network errors when reporting achievement progress to Game Center. If there was an error, it is important for your app to store any progress updates for later retransmission. This retransmission should occur at the soonest opportunity and without requiring separate user action.

Displaying Achievements

Players can view their progress on your game's achievements in the Game Center application, and you may also allow the player to view their achievements from within your game. Your game may display the default achievements screen or you may elect to retrieve achievement data programmatically from Game Center and present it using a custom interface you develop.

Be sure to clearly explain each visible achievement to the player. If the achievement is incomplete, the description should tell the player what they need to do to earn it.

Hidden Achievements

Achievements may also be hidden until you choose to reveal them. Hidden achievements could be used to keep certain elements of your game secret (until the player achieves them), or used in conjunction with add-ons made via In-App Purchase. A hidden achievement is revealed to the user the first time your application reports progress on it to Game Center.

Achievement Points

The completion of an achievement also earns the player a number of *points*. This allows your application to offer achievements with varying levels of difficulty or effort required. You may configure the number of points earned for each achievement. Each game has a maximum of 1000 points that can be earned from achievements, and no single achievement can earn more than 100 points. Your game is not required to use all 1000 points.

Be sure to plan your allocation of points to leave room for new achievements that may be added in future versions of your game, add-ons made via In-App Purchases, and so on. The point limits remain in effect in all versions of your game.

Resetting Achievements

Your app may allow the player to reset their progress on achievements and the achievements they've earned, to allow them to re-challenge themselves in a game after completing it. In general, this should always be the result of an explicit user action.

Achievements and multiple devices

A player might have the same Game Center enabled game loaded on multiple devices. If your app has reported progress on an achievement from one device, note that the same achievement progress data will be provided by Game Center on other devices where the user plays your game. However, other game state maintained by your app may not be in sync and should be handled accordingly.

Using Multi-player features

Game Center provides capabilities that make it easy for players in multiplayer games to discover each other and play together over a network. Your game can allow the user to invite their friends into a match, use auto-matching to have Game Center find other players automatically, or use a combination of both. Matches can be initiated either from within your application or from within the Game Center application.

Adding players to a multiplayer match

In the most common scenario, a player with your game already running may wish to start a multiplayer match with one or more friends. Your app can present a standard matchmaking interface allowing the user to select one or more friends and invite them into the match. Each friend invited then receives a push notification asking them if they want to join. If the friend has the game installed and accepts the invitation, your application launches on their device (if it was not already running) and processes the invitation. Once the specified number of players have joined, your app dismisses the matchmaking screen and the match can begin.

In another scenario, a player may use the built-in Game Center application to select friends and invite them into a match in your game. In this case, your app is launched automatically and provided a list of one or more players to be invited. If your game uses invites to start multiplayer matches, be sure to handle both of the scenarios outlined above.

In either case, in the event the friend being invited does not have the app installed on their device they are given the option to launch the App Store directly from the notification.

The standard interface also provides the option for the hosting player to have Game Center automatically fill remaining slots with other players available for a match. For example, if your game requires four players, a group of three friends can have Game Center fill the fourth slot (if someone else is available to play) without your application needing to do any additional work.

Your game can also initiate auto-matching programmatically. With programmatic auto-matching the player is automatically connected into a match with other players available to join the game. You are free to customize the interface the player sees while the match is formed.

At any time before the match begins, the host may elect to cancel an outstanding invite and send new ones, fill empty slots automatically with auto-matching, or cancel the match altogether. If using auto-matching, your app may query Game Center to see the flow rate of players going into matches and help the user decide whether the match they want to play is likely to be filled promptly.

Sometimes you may already have a match running and wish to add another player to it. For example, if your game needs four players and one gets disconnected, you can offer the option to find another available player instead of aborting the match in progress.

Processing invitations

Your app can process multiplayer invitations by implementing an *invitation handler*. To handle invitations promptly, it is important that your application authenticate the player and set up the invitation handler as early as possible after launching.

The invitation handler will be invoked either when the player has accepted an invitation from a friend to join a match, and should present the built-in matchmaking view to the player; or it may be invoked when your app is launched from the built-in Game Center app, and should create a new match and pre-populate the matchmaking view with the list of players the user already selected in Game Center to invite into the match.

Note that the player may receive invitations at any time. Invitations could arrive during gameplay, or while your app is suspended, or while the user is inviting other players to start a match, and so on. It is important for your app to plan for and properly handle each of these cases. It may be necessary to stop other activities in progress if the user chooses to accept the new invitation. For example, if the player accepts an invitation while another match is already in progress, your app should disconnect from that match, reset application state as needed, and process the new invitation accordingly.

Customizing matchmaking

The Game Kit API enables you to tailor auto-matching behavior to the specific requirements of your app through *player groups* and *player attributes*. Player groups allow you to specify logical groupings of players who may wish to play together. For example, in a racing game you might specify a different player group for each track, thereby enabling players who choose a particular track to only be auto-matched with others who also wish to join that same track.

Player attributes allow your game to define specific roles that different players will have in the game. You can then allow the player to select the role they wish to play. For example, in a fantasy roleplaying game, each role might represent a different class of player character with different abilities.

Communication between players

The Game Kit API provides a convenient abstraction for the network and does the work to transmit data (and voice) between the participants in the game. The API does not define the format of your network messages. Instead, it simply treats your messages as data to transmit. This provides you with great flexibility in designing your network game.

By default, Game Center creates a network of peer-to-peer connections to connect all of the participants in a match. Game Center supports a maximum of 4 players in a match configured for peer-to-peer communication.

Data transmitted by the Game Kit API can be sent *reliably*, where the match retransmits the data until it is received by the target(s), or *unreliably*, where it sends the data only once. Reliable messages should be no larger than 87K bytes in size, and unreliable messages should be no larger than 1K bytes. Despite the limits, care should be taken to use the smallest message length and to send messages at the lowest frequency that is needed.

Hosting games on your own server

Some games may want or require a server to be involved in the match. For example, your game may want its own server to arbitrate the rules of the game, or to maintain persistent data. In this case, you can use Game Center to provide a *hosted match*. Game Center supports a maximum of 16 players in a match configured for hosted communication.

Hosted matches generally involve a centralized hub used for communication. Creating a hosted match requires you to implement all of the low-level networking required for your app and to provide your own server to act as the host.

Adding Voice Chat

A great feature of Game Center is the built-in support for in-game voice chat between players. Your game can create one or more voice chat channels, each containing some or all of the players connecting to a match. When the player speaks into a channel, only the participants connected to the same channel can hear it. Optionally, additional controls may be presented to allow the user to set volume levels, mute specific players within the channel, or present an indicator for when a user starts or stops speaking. Voice chat is supported for peer-to-peer connections over wifi.

Using the Game Center service

The Game Center service (accessed via the GameKit APIs) provides player authentication, leaderboard and achievement data, and other services.

As with any network service, the Game Center service may become unreachable at any point and your app must be prepared to fall-back gracefully. Always check for network errors when authenticating players, posting scores, or updating achievements. If there was an error, it is important for your app to store any leaderboard and achievement updates for later retransmission. This retransmission should occur at the soonest opportunity without requiring separate user action.

The data maintained by the Game Center service should only be used to provide the services and functionality necessary for your application, and may not be transferred or copied to other services. For example, you may not download Game Center leaderboard data to post on your app website or to another service.

To maintain an optimal user experience, your app should only query the service for data it needs to use or display. For example, do not attempt to retrieve the entire leaderboard for your game all at once.

Using the Game Center sandbox

During development, you should test your application to ensure that Game Center features are working correctly. Apple provides a sandbox environment to allow you to test your application separately from the live version or other live apps.

As a developer, your Game Center account may be logged into either the sandbox environment for testing, or into the production environment for playing games that are already live, but not both. To switch, you must log out of your account from the built-in Game Center app, then run any Game Center enabled app and re-enter your account credentials. If that app is provisioned for development, your account will be logged into the sandbox. If that app is already live, you will be logged into the production environment.

To prevent revealing unpublished apps you or your friends are testing, information regarding your friends recently played games is not shared among accounts using the sandbox.

Other topics

Checking for Game Center support

The Game Center API and Game Center online service may only be used with devices running iOS 4.1 or later. If your app also targets operating system versions prior to iOS 4.1, you should query at run-time for the presence of Game Center APIs and for the OS version, and provide a graceful fallback if support is not indicated.

In addition, your Game Center enabled game could be loaded on a device that does not support Game Center, regardless of OS version. On such devices, player authentication will return an error indicating the current device does not support Game Center.

If Game Center is not supported on the current device, your game should be prepared to fall-back gracefully and proceed with Game Center features disabled, if it is able to do so.

If your game *requires* Game Center features in order for the game to function (for example, a multi-player game that requires Game Center to match players) then you must specify that requirement by listing `UIRequiredDeviceCapabilities` in your `info.plist` with the `gamekit` key set to `true`.

See the *Game Center Programming Guide* for more information about checking for Game Center support.

Supporting apps that are not Games

Although the Game Center feature set is designed for use in games, it may also be used in other non-game apps as well. Consider using Game Center to increase the discoverability of your app or leverage your users' friends network. Note that multi-player game invites use game-centric terminology to invite friends to "Play Now".

Using Game Center and other social networks

A number of conditions and best practices need to be followed if using Game Center and another social network in the same app.

- **Friends.** All Game Center friend connections are created through the Game Center app. There is no facility to "import" friend lists from another network. Also, you may not provide the player's friends or related data to another network service.
- **Leaderboards.** You may not download Game Center leaderboard data and provide that data to another network. You can report scores from your app both to Game Center and to another network. If your app already knows the players best scores (or can retrieve them), you can provide that data to Game Center. Note that Game Center will use the current date.

- **Achievements.** You may not download Game Center achievement data and provide that data to another network. Your app can report achievements both to Game Center and to another network.
- **Multi-player invites.** You may not use Game Center to send game invites to another network, and may not initiate a Game Center game invite from another network.
- **In-Game Voice Chat.** Using Game Center Voice Chat with another network or outside of the same Game Center app is not supported.

Presenting Game Center in your app

Be sure that your app presents Game Center functionality in a way that is integrated with the game and fits its style. Rather than using the Game Center logo as a label or providing a separate "Game Center" menu item, you should choose terms that present your use of Game Center functionality from the player's perspective. For example, you might provide menu items such as "Best scores" or "Achievements" or "Multiplayer", that leads to the appropriate data or views provided by Game Center.

Presenting Game Center in your App Store presence

You may reference Game Center in your application description but you should avoid branding your application icon with the Game Center logo. Also, it is not necessary to add Game Center to your search keywords. The App Store will add "Game Center" keywords automatically to Game Center enabled apps. This does not count towards the 100 character limit.

Supporting multiple apps

The Game Center data associated with your app is uniquely associated with the app's bundle identifier. Note that related apps such as "My App Lite" and "My App Pro" (with different bundle identifiers) each will have their own separate Game Center data.

The same Game Center data is provided to all variants of a universal application.

Nothing herein is intended to modify the iOS Developer Program License Agreement, the iOS Developer Program Enterprise License Agreement, the iOS Developer Program University Agreement, and/or the iOS Developer Program University Student License Agreement ("Agreement"), as they may be modified by Apple from time to time. In the event of any conflict or inconsistency between the Agreement and this document, the Agreement shall govern. Apple may at any time, and from time to time, with or without prior notice to You modify this document as well as any features, functionality or services described herein. You understand that any such modifications may require You to change or update Your Applications at Your own cost. Apple shall not be liable for any losses, damages or costs of any kind incurred by You or any other party arising out of or related to any modification or discontinuation of this document or any of the features, functionality or services described here.