

SNAP Mobile: Standard Game Requirements

Version 2.0; June 20, 2007

SNAP Mobile

NOKIA

Copyright © 2006, 2007 Nokia Corporation. All rights reserved.

Nokia and Forum Nokia are trademarks or registered trademarks of Nokia Corporation. Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. Other product and company names mentioned herein may be trademarks or trade names of their respective owners.

Disclaimer

The information in this document is provided “as is,” with no warranties whatsoever, including any warranty of merchantability, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample. This document is provided for informational purposes only.

Nokia Corporation disclaims all liability, including liability for infringement of any proprietary rights, relating to implementation of information presented in this document. Nokia Corporation does not warrant or represent that such use will not infringe such rights.

Nokia Corporation retains the right to make changes to this specification at any time, without notice.

License

A license is hereby granted to download and print a copy of this specification for personal use only. No other license to any other intellectual property rights is granted herein.

Contents

1	Introduction	5
1.1	Audience.....	5
1.2	Definitions	5
1.3	Document organization	5
2	Mandatory Requirements and Best Practices	6
2.1	Coding practice.....	6
2.2	Testing practice.....	6
2.3	User interface.....	6
2.4	Registration and login	7
2.5	Instant Messaging and Presence Services.....	8
2.6	Multiplayer.....	9
2.7	Ranking.....	10
2.8	Connectivity.....	11
3	Terms and abbreviations	12
4	Further reading	14
5	Evaluate this resource	15

Change History

March 20, 2006	Version 1.0	Initial document release at www.forum.nokia.com
June 20, 2007	Version 2.0	Important changes to requirements; 006, 009, 010, 012, 019, 020, 023, 024, 027, 032, 033, 034, 035. Important changes to best practices; 002, 003, 008, 015, 020.

1 Introduction

This document is a part of Nokia's Scalable Network Applications Platform (SNAP) Mobile SDK. It lists the technical, usability, and publishing requirements that apply to SNAP Mobile games. This document is used as a standard reference for SNAP Mobile game compliance testing.

1.1 Audience

This guide is directed to game developers who use the SNAP Mobile Client API to develop SNAP Mobile games for mobile devices that support the Java™ Platform, Micro Edition (Java™ ME) technology and who intend to include connected or multiplayer features in these mobile games. The guide assumes that you are familiar with standard online game production and development principles.

1.2 Definitions

The following definitions are important to keep in mind when considering the difference between mandatory requirements and best practices.

- **MANDATORY REQUIREMENT** is specified with the word **shall**. Failure to comply with a mandatory requirement will result in the game being rejected from SNAP Mobile compliance testing.
- **BEST PRACTICE** is specified with the word **should**. Making use of recommended best practices will enhance the user experience and may also make things go smoother during game development and testing.

Note: If any of the requirements or best practices specified in this document conflict with a feature or functionality that developers want to implement in their games due to a valid business or design reason, then the developers should contact their SNAP Mobile Development Contact¹ to find a solution.

1.3 Document organization

Mandatory Requirements and Best Practices are organized into the following eight different categories:

- **Coding Practice** — Describes fundamental coding practice.
- **Testing Practice** — Describes fundamental testing practice.
- **User Interface** — Focuses on user interface and user experience to make the connected experience consistent and easy to users.
- **Registration and Login** — Lists basic login and account creation rules.
- **Instant Messaging and Presence Services** — Implements fully functional features such as the Friends List, Presence, and Chat/IM.
- **Multiplayer** — Implements fully functional features such as Lobbies, Game Rooms, and Matchmaking.
- **Ranking** — Lists basic ranking rules.
- **Connectivity** — Demonstrates communication capability over a network correctly.

¹ The SNAP Mobile Development Contact referred to in this document can be provided via different channels – that is, through a SNAP Mobile business development representative or through Forum Nokia – once the developer has access to the SNAP Mobile development environment.

2 Mandatory Requirements and Best Practices

This chapter provides Mandatory Requirements and Best Practices for developing a SNAP Mobile game.

2.1 Coding practice

Mandatory Requirement

- S2.0/001 The game shall only use the SNAP Mobile servers and services. Use of any other servers or services outside the SNAP Mobile platform by the game is prohibited.
- S2.0/002 All SNAP Mobile games shall use their own unique Game Class ID obtained from the SNAP Mobile Development Contact.
- S2.0/003 The game shall be compiled using a supported version of the SNAP Mobile Client API. This means that once the game begins compliance testing, it shall use the latest released version of the SNAP Mobile Client API or the version just prior to that one.
- S2.0/004 The game shall process SNAP Mobile error and status codes and take appropriate actions to handle them, as outlined in the *Synchronous response data* and the *Asynchronous response data* appendixes in the *SNAP Mobile: Game Developer's Guide for Java™ ME Clients* document. This shall include providing the user with the appropriate error message as outlined in these appendixes.
- S2.0/005 Mandatory JAD and JAR file parameters, specified in *Appendix D: JAD and JAR file requirements* of the *SNAP Mobile: Game Developer's Guide for Java™ ME Clients* document, shall be present in the game's JAD file and its JAR file manifest.
- S2.0/006 The game JAD and JAR file parameters shall follow code practices specified in *Appendix D: JAD and JAR file requirements* in the *SNAP Mobile: Game Developer's Guide for Java™ ME Clients* document.

Best Practice

- B2.0/001 The game should not permanently change any device settings. Before exiting, the game should restore any altered device settings to their original values.
- B2.0/002 Developers should follow the best practices outlined in the *Software development best practices* section in the *SNAP Mobile: Game Developer's Guide for Java™ ME Clients* document.

2.2 Testing practice

Best Practice

- B2.0/003 Developers should follow the best practices in the *Testing guidelines* section of the *SNAP Mobile: Game Developer's Guide for Java™ ME Clients* document.

2.3 User interface

Mandatory Requirement

- S2.0/007 The game shall include "GO ONLINE" in the main menu to gain access to the SNAP Mobile community services.

S2.0/008 When in online mode, the game shall use the terminology listed in Table 1 (case insensitive). This requirement only applies to English language games.

Terminology	Functionality
CREATE ACCOUNT	Used in the account creation screen.
LOGIN	Used in the login screen.
USER NAME	Meaning a unique user ID required for online access.
PASSWORD	Meaning a secured word to verify user identity.
FRIENDS	Used in the online menu. Meaning the list of friends (for example, "Friends List") that users create for the purpose of messaging and handling gameplay challenges.
MESSAGES	Used in the friends screen. Meaning messages to be exchanged between friends.
RANKINGS	Used in the online menu. Meaning the list of ranking scores.
LOG OUT	Used to end the online connection.

Table 1: Terminology to be used in online mode

- S2.0/009 When the game is launched, the game shall display on the game title screen for a minimum of two (2) seconds the following text: 1) centered at the top of the screen: "Powered by Nokia's SNAP Mobile"; 2) centered at the bottom of the screen: "snaparcade.com". All text shall be of the same font type, color (preferably black or white), style, and size. The user shall have no option of skipping this title screen.
- S2.0/010 The game shall display "Games, friends and lots of fun! snaparcade.com" on a page within two (2) clicks from the main menu (for example, HELP, ABOUT, CREDIT).
- S2.0/011 When a list of items such as menu choices cannot be fully displayed in one screen, indicators such as arrows or scroll bars shall show the availability of other items.
- S2.0/012 An animation shall inform the player of the progress of an operation requiring connectivity that takes 5 seconds or longer such as login, logout, rankings, upload, download, or matchmaking.

Best Practice

- B2.0/004 The game should use the player's SNAP Mobile user name as a player identifier throughout the game.

2.4 Registration and login

Mandatory Requirement

- S2.0/013 The game shall use the SNAP Mobile Authentication.
- S2.0/014 The game shall allow the user to create an account and the account creation process shall adhere strictly to the functionality and flow used in the reference implementation as outlined in the *SNAP Mobile: Registration and Login Guidelines for Java™ ME Clients* document.
- S2.0/015 The user shall receive a response within sixty (60) seconds of submitting a login request.
- S2.0/016 For user names, the game shall only allow the player to enter four (4) to fifteen (15) characters from the set "a" - "z", "A" - "Z", "0" - "9", "_" (underscore), and "-" (hyphen).

- S2.0/017 For the password, the game shall only allow the player to enter four (4) to fifteen (15) characters, consisting of the 7-bit ASCII character codes between character codes 32 (“ ”) and 126 (“~”), inclusive.
- S2.0/018 The password shall be masked and not be displayed in clear text except when the user inputs the password into the game (for example, when creating a user account).
- S2.0/019 The game shall allow the players to save their user name and password on the mobile device so they can log in again without retyping their user name and password. To do so, a “Save my login information” style check box shall be included; if the check box is cleared, the game shall clear the locally stored user name and password and the default setting of the check box shall be unchecked.
- S2.0/020 Since passwords are case sensitive, users shall be able to see the case of each character they are using (either upper or lower case) while they are entering their password.

Best Practice

- B2.0/005 The user should receive a response within thirty (30) seconds after submitting a login request.

2.5 Instant Messaging and Presence Services

Mandatory Requirement

- S2.0/021 The game shall provide the user with the ability to manage the friends list (that is, to add, remove, accept, or reject friends).
- S2.0/022 The game shall prompt the users to confirm their request prior to permanently removing a friend from the friends list.
- S2.0/023 The game shall provide the user with the ability to see the current online status of friends on the user’s friends list. The online status shall be displayed as either (a) “online”, (b) “busy”, or (c) “offline” according to the cases outlined in the *Presence states vs. online states* section of the *SNAP Mobile: Game Developer’s Guide for Java™ ME Clients* document. Alternatively, the game shall display color or graphics as a substitute for the specified text so long as they are consistent with the same cases and easily understood.
- S2.0/024 The game shall provide the user with the ability to exchange chat messages with any friend whose online status is “online”. Additionally, the game shall allow the user to send chat messages to a friend whose online status is “busy” only in two cases: (a) the developer has chosen to allow messaging during gameplay and the “busy” friend is in the same GCID; or (b) the “busy” friend in another GCID initiated the chat.
- S2.0/025 All messages using the IMPS services between “friends” in different games shall contain only chat messages between users and shall not contain metadata.
- S2.0/026 If the game offers a challenge mode, the game shall only allow the user to challenge a friend who is in the same game and whose online status is “online”.
- S2.0/027 If the game allows the users to set their online status, there shall be only two options: (a) “online”, or (b) “busy”. Alternatively, the game shall display color or graphics as a substitute for the specified text so long as they are consistent with the same cases and easily understood.

Best Practice

- B2.0/006 The game should queue messages received from outside of a game a user is playing while the user is in a game session.
- B2.0/007 If the game offers a challenge mode, the game should provide the user with the ability to challenge a friend in the same game from the friends list.
- B2.0/008 If the game allows the users to self select their online status to “Busy”, they should keep the ability to send messages to any online user that they could message before changing their online status.

2.6 Multiplayer

While SNAP Mobile games are required to have community functionality (such as, login and registration, rankings, messaging and presence functionality), multiplayer functionality may be optional. This section’s requirements only apply to games with multiplayer functionality.

Mandatory Requirement

- S2.0/028 By default, the game shall not attempt to create a game lobby. If the game design requires the use of dynamic lobby creation from within the game, the developers shall contact their SNAP Mobile Development Contact.
- S2.0/029 Lobbies created for the purpose of matchmaking through random start, sort start, or challenge start modes shall not appear to users. See the *Working with lobbies and game rooms* section in the *SNAP Mobile: Game Developer’s Guide for Java™ ME Clients* document.
- S2.0/030 The game shall not use more than one matchmaking mode in a given lobby.
- S2.0/031 Any game that starts through the challenge start or random start modes shall allow its players to cancel the matchmaking request.
- S2.0/032 If a synchronous game allows a player to pause gameplay, the game shall present that player a pause menu that allows him/her to resume gameplay. Only the player who pauses the game shall be able to resume gameplay.

Note: The definitions for a synchronous game and asynchronous game in Chapter 3, “Terms and abbreviations,” shall apply to this requirement and each of the following requirements that uses these terms.

- S2.0/033 In a synchronous game, if any player pauses the game, gameplay shall be paused for all players. When this happens, the game shall inform all player(s) about who has paused the game.
- S2.0/034 In a synchronous game, when a player quits the game, the other player(s) shall be informed that the game has ended and who has quit the game.
- S2.0/035 In an asynchronous game, when a player quits the game, the other players shall be informed that the player is no longer in the game.
- S2.0/036 At the end of a successful game session, all of the players in the game shall be returned to the same navigation point from which the game was launched.

Best Practice

- B2.0/009 If applicable, the game should implement at least two of the matchmaking modes outlined in the Matchmaking section in the *SNAP Mobile: Game Developer's Guide for Java™ ME Clients*. One of these matchmaking modes should be Random Mode.
- B2.0/010 If the number of players in the game session drops below the minimum number required for the game, the game session should either terminate or replace the missing players.
- B2.0/011 Data packet efficiency should be considered when designing and developing the game.
- B2.0/012 When using the in-game message function, the game should limit the number of people in one game room to less than twenty (20) people to avoid unnecessary message broadcasting.
- B2.0/013 If the game uses custom event masking, follow the guidelines for this outlined in the Event Masking section in the *SNAP Mobile: Game Developer's Guide for Java™ ME Clients* to avoid performance issues.
- B2.0/014 In an asynchronous game, when a player pauses the game, the game should present that player with a pause menu with the option to resume gameplay, and the game should inform the other players that the player who paused has become inactive while they may continue gameplay.
- B2.0/015 In a synchronous game, when a player pauses the game, the other player should be given the option to exit the game if it is not restarted after a reasonable amount of time.
- B2.0/016 If users lose their connection for more than the server-defined time-out of ninety (90) seconds, this should be handled as if they voluntarily quit the game session from the point of view of the other players.

2.7 Ranking

Mandatory Requirement

- S2.0/037 The game shall submit the score after each game played, regardless of the score being better or worse than the user's personal best.
Note: This requirement only applies to game modes that support score submission.
- S2.0/038 It shall be possible to successfully submit a score earned for a specific game session only once.
- S2.0/039 If a score submission fails for any reason, the game shall resubmit that score until score upload is successful.

Best Practice

- B2.0/017 To participate in SNAP Mobile tournaments, the game should not submit cumulative scores, but rather the score for any given game session.
- B2.0/018 The game should submit as much detail as possible, beyond the grand total of points scored. The game should provide a score in as many measurable categories as possible that are statistically relevant to the game.

2.8 Connectivity

Mandatory Requirement

S2.0/040 The game shall return its player to the main menu when one of the following events occurs:

- if a fatal error is received from the server, or
- If the game loses its connection and cannot restore it within ninety (90) seconds – This includes, but is not limited to, external incoming communication such as voice call.

S2.0/041 The game shall close all open network connections and log out from SNAP Mobile community services when the user exits the game.

Best Practice

B2.0/019 When the game times out after a reasonable interval of trying unsuccessfully to establish a server connection, it should display a notification message “Network error may have occurred. Check access point settings and/or signal strength.” to the user.

B2.0/020 When the game receives a “989 - transport timeout warning” message, it should send a notification message such as “Slow network connection”, but the game should not interrupt network processing.

3 Terms and abbreviations

Term or abbreviation	Meaning
API	Application programming interface. The SNAP Mobile Client API is a client library that provides networking, community services, and graphical user interface functionality to the game.
Asynchronous Game	A game that is coded in such a way that one player can compete with other players, but does not wait for them to take their turn or make their moves. Typically this is like a racing game.
Community Services	Web, IMPS, and SNAP Game services that support SNAP Mobile requests.
Challenge Mode	Matchmaking mode in which one player requests a specific opponent.
Game Class	Base unit of a network multiplayer game. Game classes are typically implemented across a cluster of servers, with each cluster able to support multiple game classes. Game classes contain three discrete components: game lobbies, game rooms, and users.
Game Class ID	A game class ID serves as a global identifier for a game class. In practice there is a one-to-one correspondence between a game class and a game.
Game Room	A dynamic object created in a lobby. From the perspective of a player, a game room is a place to meet and engage in gameplay.
Game session	The period from game start until the game stops.
IMPS Services	Community services that provide instant messaging, presence, and the management of a listing of friends.
Join Mode	Matchmaking mode in which a player joins a specified game room and lobby.
Lobby	A static object that can hold multiple Game Rooms. Lobbies exist until explicitly deleted.
Matchmaking Mode	One of four ways of matching players in a game: Challenge, Join, Random Start, and Sort Start modes.
Message	A text message between SNAP Mobile users, sometimes referred to as "chat" or "instant messaging."
Presence	IMPS functionality that indicates whether a SNAP Mobile client is available for instant messaging or gameplay.
Random Start Mode	Matchmaking mode in which a user joins an available game room in Random Lobby to play against any other users who enter that room.
Sort Start Mode	Matchmaking mode in which a user joins an available game room in the Sort Lobby. Users are joined to a game room to either top off a room or load balance across available rooms.
SNAP Mobile Authentication	Community service that securely identifies users of online games based on their user IDs and passwords. It uses policy-driven rules and custom properties to regulate user access to games and other services.
SNAP Communications	A custom network transport system that supports SNAP Game Services. SNAP Communications is optimized for online multiplayer gameplay.
SNAP Game Services	Community services that provide network multiplayer functionality for SNAP Mobile games and game rooms.

Term or abbreviation	Meaning
SNAP Mobile Client	Client terminal, handset, phone, or other device that uses the SNAP Mobile Client API to communicate through the SNAP Mobile gateway to the community services.
SNAP Mobile Gateway	Session manager and virtual client system that handles requests from SNAP Mobile clients and responses from community services.
SDK	Software development kit
SNAP Mobile Game	A game that both provides access to the SNAP Mobile Community and provides one or more SNAP Mobile in-game functions.
Synchronous Game	A game which is coded in such a way that one player cannot make a move without the other player taking his/her turn. Typically this is like chess or poker.
User	Participant who accesses the community services through a SNAP Mobile client.
Web Services	Community services that provide user account and profile management, the rankings engine, event reporting functionality, and user authentication; all Web service data is centralized for use by games.

4 Further reading

The SNAP Mobile Client SDK offers documentation and resources to support SNAP Mobile game development. To obtain the latest SDK, go to <http://www.forum.nokia.com/games/snapmobile/>. The SDK includes the following documentation:

- SNAP Mobile Client API Javadocs
- SNAP Mobile: Standard Game Requirements
- SNAP Mobile: API Compatibility Test Instructions
- SNAP Mobile: Game Developer's Guide for Java™ ME Clients
- SNAP Mobile: Registration and Login Guidelines for Java™ ME Clients
- SNAP Mobile: Ranking Guide for Java™ ME Game Developers
- SNAP Mobile: Emulation Environment User's Guide
- SNAP Mobile: Hello World Tutorial for Java™ ME Clients
- SNAP Mobile: Sample Game Users Guide

SNAP Mobile also partners with Sun Microsystems, Inc. to offer SNAP Mobile resources through the Sun Java™ Wireless Toolkit (formerly the J2ME™ Wireless Toolkit). These resources include:

- SNAP Mobile Sample Game

This user application is packaged with sample code that demonstrates common uses of the SNAP Mobile Client API; for example, logging in to the online community, initiating multiplayer game sessions, messaging, retrieval of rankings and game scores, and so on. The toolkit is available at <http://java.sun.com/products/sjwtoolkit/>.

5 Evaluate this resource

Please spare a moment to help us improve documentation quality and recognize the resources you find most valuable, by [rating this resource](#).