

# SNAP Mobile: Server Administration Guide for Game Developers

Version 1.0; July 10, 2008

# SNAP Mobile

# NOKIA

Copyright © 2008 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 document at any time, without notice.

#### **License**

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

## Contents

|          |                                                           |           |
|----------|-----------------------------------------------------------|-----------|
| <b>1</b> | <b>Introduction.....</b>                                  | <b>5</b>  |
| 1.1      | Scope and audience.....                                   | 5         |
| 1.2      | Using this document.....                                  | 5         |
| <b>2</b> | <b>Accessing the CSR tool.....</b>                        | <b>6</b>  |
| <b>3</b> | <b>Managing users.....</b>                                | <b>8</b>  |
| <b>4</b> | <b>Managing assets.....</b>                               | <b>10</b> |
| 4.1      | Creating an asset class for a game.....                   | 10        |
| 4.2      | Managing asset classes.....                               | 12        |
| 4.2.1    | Managing the metadata properties for an asset class.....  | 13        |
| 4.2.2    | Managing the asset processors used by an asset class..... | 14        |
| 4.3      | Uploading an asset into an asset class.....               | 14        |
| 4.4      | Managing uploaded assets.....                             | 16        |
| <b>5</b> | <b>Managing rankings.....</b>                             | <b>19</b> |
| 5.1      | Setting up a Highscore board for a game.....              | 19        |
| 5.2      | Setting up a High Score Filter board for a game.....      | 21        |
|          | <b>Appendix: Locale identifiers.....</b>                  | <b>23</b> |
|          | <b>Evaluate this resource.....</b>                        | <b>26</b> |

## Change history

|               |             |                                                                                                                                                         |
|---------------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| July 10, 2008 | Version 1.0 | Initial document release. Replaces CSR tool procedures in SNAP Mobile rankings and asset management documentation for the version 1.x SNAP Mobile SDKs. |
|               |             |                                                                                                                                                         |

# 1 Introduction

This guide provides instructions for configuring the SNAP (Scalable Network Application Package) Mobile Development server environment through the CSR tool. The CSR Tool is a Web-based administrative tool that allows game developers to perform the server configurations that are required to support the asset management and ranking functionality in the SNAP Mobile API. This functionality relates to the following SNAP Mobile features:

- Methods in the `Session` class for reporting game statistics (such as scores) and retrieving player rankings that are based on those statistics: `reportScores (...)`, `getProximityRankings (...)`, `getPlayerStatistics (...)`, `getTopRankings (...)`
- Methods in the `AssetManager` class for uploading and retrieving assets: `fetchAsset (...)`, `listAssets (...)`, and `uploadAsset (...)`

**Note:** The classes were introduced in the 2.0 version of the SNAP Mobile Client SDK. There are similar methods in the 1.x versions of the SDK. The CSR tool functionality is compatible with ranking stats, filters, assets, asset classes, and related features for these earlier versions of the SDK.

This guide replaces the CSR tool configuration procedures included in the following 1.x SNAP Mobile SDK documents that are no longer available:

- SNAP Mobile: Ranking Guide for Java ME Game Developers
- SNAP Mobile: Asset Management Guide for Game Developers

## 1.1 Scope and audience

This document is intended for game developers who are using the SNAP Mobile SDK to implement rankings and asset management functionality in a multiplayer game.

## 1.2 Using this document

This document is divided into the following sections:

- Chapter 2, “Accessing the CSR tool,” explains how developers gain access to this tool.
- Chapter 3, “Managing users,” explains how to create test users for a game.
- Chapter 4, “Managing assets,” explains how to create and modifying assets, asset classes, and their properties.
- Chapter 5, “Managing rankings,” explains how to set up ranking statistics, filters, and other features that support player rankings.

## 2 Accessing the CSR tool

This chapter explains how developers gain access to this tool.

To log in to the CSR tool:

1. In a Web browser, open the Login screen for the CSR tool by using the URL provided in the SNAP Mobile Development Server Access package.

**Note:** To receive a CSR tool username and password, it is necessary to purchase the Live SNAP Mobile Development Server Access package from the [e-store](#) (requires login to Forum Nokia). The e-store is accessible from the [SNAP Mobile Technology page](#). If you do not see a link to the package, navigate through the e-store pages to “SnapMobile Development Server Access,” or, if you have a Forum Nokia Pro account, to “SnapMobile PRO Development Server Access.”

Note that you cannot use this tool without development server access. Access to the CSR tool and the development server is not required to complete game development. If you do not have access to the CSR tool, you can use the locally hosted Emulation Environment that is part of the SNAP Mobile SDK to set up simulated versions of the asset management and ranking features described in this document, and then test and pre-certify your game against them.

2. In the Login screen (shown in Figure 1), type the user name and password provided for this tool.

Figure 1: Login screen of the CSR tool.

3. Click the Login button to open the main screen of the CSR tool. See Figure 2 for an example of this screen.

Figure 2: Main screen of the CSR tool.

The CSR tool supports the following functionality for game developers:

- **Users** - For creating test users for a game. See Chapter 3, “Managing users.”
- **Asset Management** – For creating and modifying assets, asset classes, and their properties. See Chapter 4, “Managing assets.”
- **Rankings** - For setting up ranking statistics, filters, and other features that support player rankings. See Chapter 5, “Managing rankings.”

### 3 Managing users

This chapter explains how to create test users for your game through the CSR tool. These test accounts provide users on the Development server for your game. To test interactive processes in a connected, multiplayer game, you often need to log in to the Development server through the game with more than one user. The CSR tool allows you to create as many as you need.

To create a user:

1. From the main screen of the CSR tool, click the **Users tab**.
2. Click **Add User** from the Action Items menu on the left of the screen to open the Add User screen. (See Figure 4 for an example of the Action Items menu.)
3. Specify user information in the Add User screen. Fields with an asterisk (\*) require data.

**Add User** ✓ SUBMIT ✗ REJECT

**Email**

Email Address  
myaddress@mine.com

Confirm Email Address  
myaddress@mine.com

Require Email Validation  
 Y  N

**Security Question**

Security Question\*  
Name of your first school?

Security Answer\*  
ZvxqGbAZ16T (randomize)

**Personal**

First Name\*  
myNewuser

Last Name\*  
myUsersLastName

Date of Birth\*  
10 / 10 / 1980

Region\*  
United States

Language\*  
English

**Account**

Username\*  
newUser

Password\*  
xxxxxxx

Confirm Password\*  
xxxxxxx

Figure 3: Add User screen

- Select **N** for the Require Email Validation option.

- Select a security question and type an answer (or click **Randomize** to create a randomly generated security answer). Note that there is no mechanism in place to retrieve your password based on the security question and answer. Therefore, the Randomize option is always appropriate.
  - Type your personal information, entering your date of birth in the following format: MM/DD/YYYY.
  - Select the region and language that best apply to you or your test user.
  - Your user name must be unique.
4. Click the **Submit** button to enter your user information and receive the confirmation notice shown in Figure 4.

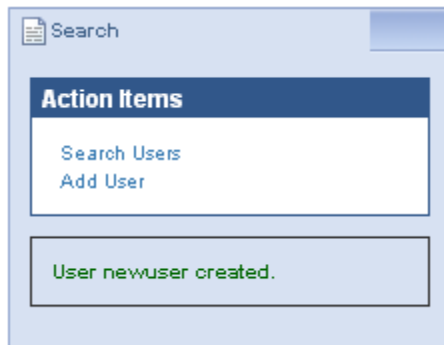


Figure 4: Confirmation notice: user created.

## 4 Managing assets

This chapter provides information on how to configure assets and asset classes for a game. This functionality supports the asset management methods described in the Introduction.



Figure 5: Asset Management features.

The Asset Management tab supports the following features:

- **Game Configuration tab:** This tab provides a way to create and assign asset classes to a game, upload a new asset to an asset class, and manage asset properties such as asset metadata. However, do not use the Processors sub-tab unless developer support directs you to do so.
- **Uploaded Assets tab:** This tab provides a mechanism to search for assets, view their properties, and change the validation status of an asset, which affects whether the asset is available to a game.
- **Asset Classes tab:** In most cases, you do not need to use this functionality. This tab supports the configuration of asset classes that can be used across multiple game class IDs (GCIDs). In general, you should use only the asset classes that you create for your game. As a consequence, you should not use this tab unless you are creating more than one game, and in that case, you should only use asset classes that are created for your games.
- **Processors tab:** In most cases, you do not need to use this functionality. This tab supports the server-side applications that implement rules for processing of assets. Note that new asset processor creation requires server development and therefore requires direct contact with a SNAP Mobile developer contact.

**Note:** This guide does not describe features in the Asset Classes or Processors tabs, nor does it describe asset processor management, because game developers do not need to use these features unless instructed to do so by SNAP Mobile developer support.

### 4.1 Creating an asset class for a game

The following explains how to create an asset class for your game. Asset classes specify the default properties of their assets. Note that all assets belong to an asset class.

**Note:** It is necessary to create at least one asset class before using the asset management functionality in a game. Asset upload and download methods always specify an asset class.

To create an asset class for a game:

1. From the main screen of the CSR tool, click the **Asset Management tab**.
2. Click the link to your game (located in the Game Configuration table). See Figure 5 for an example that includes two games (Any Game and RoShamBo).
3. Click **Add Asset Class** from the Action Items menu on the left of the screen.

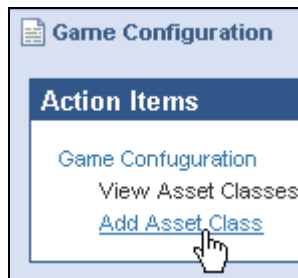


Figure 6: Adding an asset class.

4. Select **New Asset Class** from the top drop-down list on the screen, and then complete the remaining fields. See Figure 7 for an example followed by descriptions of each field and recommendations on how to use them. (Note that the name and description apply to the asset class, and the remaining fields specify the default properties of all new assets in the class.)

 A screenshot of a web form titled "Add Asset Class for game Any Game". The form has a blue header with a "SUBMIT" button on the right. The form contains several fields:
 

- "Choose from available assets": A dropdown menu with "New asset class" selected.
- "Name": A text input field containing "Zavatars".
- "Description": A text input field containing "English Language Avatars".
- "Locale": A text input field containing "10".
- "File extension": A text input field containing "avtrs".
- "Query Status": Two radio buttons. The first is selected and labeled "Allow filtering uploaded assets by this class". The second is labeled "Do not allow filtering uploaded assets by this class".
- "Default Validation Status": A dropdown menu with "Passed" selected.
- "Permissions": A dropdown menu with "Allow Upload & Download" selected.
- "Send Meta-data in response": A dropdown menu with "yes" selected.
- "Send billing data in response": A dropdown menu with "No" selected.
- "Meta-data is XML?": A dropdown menu with "yes" selected.

Figure 7: Creating an asset class.

- **Choose from available assets:** Select **New asset class**. To avoid future errors, do not select an existing asset class from this menu unless instructed to do so by SNAP Mobile developer support. Note that this drop-down menu pertains to asset classes, not to assets.
- **Name:** Type a name for your asset class. In general, it is a good practice to provide a unique, easily identifiable name for the asset class.
- **Description:** Type a brief description of the asset class. One use case for this field is to associate the description with a locale, so, for example, a description of one asset class might be in French, whereas the description of another might be in English, and another might be in Japanese. Then you can select the appropriate locale for a given description.
- **Locale:** Type a locale identifier, for example, 10 for English to designate that the asset class supports assets for English-speaking players. Another asset class might support localized assets for French-speaking or Mandarin-speaking gamers. Locale IDs correspond to Symbian locale identifiers. For a list of identifiers, see "Appendix: Locale identifiers." Note that the asset retrieval methods in the SNAP Mobile Client API do not allow you to specify the locale; however, because each `asset` object contains a locale field, you can iterate over a retrieved list of assets to find assets that support a particular locale.

- **File Extension:** Type a file extension for your assets, for example, `avtrs`. The extension value does not need to specify a standard file extension.
- **Query Status:** If the assets in this class use metadata in XML format, click the radio button that allows filtering. If not, do not allow filtering. Filtering is a way for retrieving an asset list based on a metadata property of assets in a given asset class. See Section 4.3, “Uploading an asset into an asset class”, for more information on asset metadata.
- **Default Validation Status:** Select **Passed** to avoid future errors. The game client can retrieve only those assets for which the validation status is set to **Passed**. The game cannot retrieve **Failed** or **Pending** assets. Note that it is possible to change the validation status of a particular asset after it has been uploaded through the **Uploaded Assets** tab.
- **Permissions:** Select **Allow Upload & Download** to avoid future errors. Note that the game should only allow uploads and downloads of assets that are appropriate to and needed by the game. Allowing end users to upload and download images and assets that they create can lead to the distribution of inappropriate content through the game because these assets are not monitored in the production environment. Also note that it is possible to upload assets to an asset class through the CSR tool (as opposed to the game) regardless of the permissions on the asset class. So, for example, if the asset class is set to **Allow Download Only** or is set to **None**, then the game cannot upload assets to this asset class, but CSR tool does allow you to upload assets to this class (see Section 4.3, “Uploading an asset into an asset class”).
- **Send metadata in Response:** Select **Yes** if this class of assets will use the asset metadata, **No** if not. Metadata can be any string that is allowed by the SGR. See Section 4.3, “Uploading an asset into an asset class”, for more on asset metadata.
- **Send Billing data in Response:** Choose **No**. This functionality is not supported.
- **Metadata is XML?:** Select **Yes** if the assets in this class use metadata in XML format. See Section 4.3, “Uploading an asset into an asset class”, for more on asset metadata.

**Note:** It is also possible to use this screen to support the use of an existing asset class by your game. However, in general, it is a good practice to use this option only if the existing asset class was created for your game.

5. Click the **Submit** button.

## 4.2 Managing asset classes

After creating an asset class, you can search for it and edit its properties by following the procedure in this section. You can also delete an asset class if necessary. Note that modifications to an asset class will affect only those assets that you upload to the class after changing the class. The changes will not affect existing assets in that class. For this reason, it is generally a good practice to avoid changing the properties of an asset class, if possible.

To manage an asset class:

1. From the main screen of the CSR tool, click the **Asset Management** tab.
2. Click the link to your game (located in the Game Configuration table) to open the list of Asset classes for the game.
3. Click the link to an asset class for your game (located in the Assets Class table). See Figure 8 for an example.

| Asset Classes in use for Any Game |                         |                             |           |                           |                   |                            |                                |                   | DELETE            |
|-----------------------------------|-------------------------|-----------------------------|-----------|---------------------------|-------------------|----------------------------|--------------------------------|-------------------|-------------------|
| <input type="checkbox"/>          | ASSET CLASS             | DESCRIPTION                 | EXTENSION | DEFAULT VALIDATION STATUS | PERMISSIONS       | SEND META-DATA IN RESPONSE | SEND BILLING DATA IN RESPONSE? | META-DATA IS XML? | PROCESSORS        |
| <input type="checkbox"/>          | <a href="#">U Test2</a> | any game sample asset class | .tst      | P                         | Upload & Download | Yes                        | No                             | Yes               | Manage Processors |

Figure 8: Selecting an asset class.

- Update the settings that apply to new assets in this class (only if necessary). For guidance, see step 4 in Section 4.1, “Creating an asset class for a game.” (As stated previously, the new properties apply to the assets that you upload after updating the class but will not affect existing assets in that class.)

**Edit Asset Class 'Test2' for Game Any Game** ✓ SUBMIT ✗ REJECT

Default Validation Status:

Permissions:

Send Meta-data in response:

Send billing data in response:

Meta-data is XML?:

Figure 9: Editing an asset class.

- Click the **Submit** button. See Figure 9, for an example of the Submit button.

**Note:** It is also possible to delete a selected asset class. See Figure 10 for an example.

|                                     | ASSET CLASS | DESCRIPTION                 | EXTENSION | DEFAULT VALIDATION STATUS | PERMISSIONS       | SEND META-DATA IN RESPONSE | SEND BILLING DATA IN RESPONSE? | META-DATA IS XML? | PROCESSORS        |
|-------------------------------------|-------------|-----------------------------|-----------|---------------------------|-------------------|----------------------------|--------------------------------|-------------------|-------------------|
| <input checked="" type="checkbox"/> | U Test2     | any game sample asset glass | .tst      | P                         | Upload & Download | Yes                        | No                             | Yes               | Manage Processors |

Figure 10: Deleting an asset class.

#### 4.2.1 Managing the metadata properties for an asset class

This section provides a procedure for viewing, adding, and deleting metadata properties for assets in an asset class. Note that this functionality is available to asset classes that were created to handle metadata in XML format that allows filtering (see Section 4.1, “Creating an asset class for a game”).

To manage metadata properties:

- From the main screen of the CSR tool, click the **Asset Management tab**.
- Click the link to your game (located in the Game Configuration table).
- Click the link to an asset class for your game (located in the Assets Class table).
- Click **Configure Meta Data Properties** from the Action Item menu to the left of the screen.
- Manage the metadata properties:
  - Deleting:** You can select and then delete a metadata property as shown in Figure 11.

| DELETE                              | PROP NAME | PROP TYPE |
|-------------------------------------|-----------|-----------|
| <input checked="" type="checkbox"/> | tea       | Integer   |

Figure 11: Deleting a metadata property.

- Creating:** You can add one or more new metadata properties by clicking the Add Properties link and then creating new properties, as shown in Figure 12 and Figure 13.


| Meta Data Props for Asset Class 'Test2' for Game Any Game |           |  |
|-----------------------------------------------------------|-----------|-------------------------------------------------------------------------------------|
| PROP NAME                                                 | PROP TYPE |                                                                                     |
| <input type="text" value="myWizard"/>                     | String    |                                                                                     |
| <input type="text"/>                                      | String    |                                                                                     |

Figure 12: Creating a metadata property.


| Meta Data Props for Asset Class 'Test2' for Game Any Game |           |  |
|-----------------------------------------------------------|-----------|-------------------------------------------------------------------------------------|
| DELETE                                                    | PROP NAME | PROP TYPE                                                                           |
| <input type="checkbox"/>                                  | myWizard  | String                                                                              |

Figure 13: Viewing the new metadata property.

- **Editing:** You can change the name and/or data type of the property by clicking the metadata property and then modifying it, as shown in Figure 14 and Figure 15.


| Meta Data Props for Asset Class 'Test2' for Game Any Game |                                             |  |
|-----------------------------------------------------------|---------------------------------------------|-------------------------------------------------------------------------------------|
| Prop Name                                                 | <input type="text" value="myWizardAvatar"/> |                                                                                     |
| Prop Type                                                 | Integer                                     |                                                                                     |

Figure 14: Editing a metadata property.


| Meta Data Props for Asset Class 'Test2' for Game Any Game |                |  |
|-----------------------------------------------------------|----------------|--------------------------------------------------------------------------------------|
| DELETE                                                    | PROP NAME      | PROP TYPE                                                                            |
| <input type="checkbox"/>                                  | myWizardAvatar | Integer                                                                              |

Figure 15: Viewing the updated metadata property.

#### 4.2.2 Managing the asset processors used by an asset class

In general, you should not use the asset processors unless the default asset management system is not sufficient. By default, the system retains all assets that you upload to an asset class, and the client can retrieve any assets in that class by listing all the assets in the class or by fetching a particular asset. You can contact a SNAP Mobile developer support representative if you have special asset processing needs. If you do need to manage asset processors, a Manage Processors link associated with each asset class provides access to an edit screen. Figure 8 provides an example of this link for the Test2 asset class.

#### 4.3 Uploading an asset into an asset class

It is possible to use the CSR tool to upload assets into an asset class. This section provides a procedure for this process. Note that it is also possible to upload an asset to an asset class through a request to the `AssetManager.uploadAsset (...)` method.

**Note:** 30720 bytes is the maximum size allowed for a single asset upload. However, you need to determine a reasonable size for the game's assets based on the network or networks that the game will run on. For example, a very high-speed network may be able to handle uploads and downloads of 30720 byte asset efficiently, but a slower network may not.

To upload an asset to an asset class:

1. From the main screen of the CSR tool, click the **Asset Management tab**.
2. Click the link to your game (located in the Game Configuration table).

- Click the “U” to the left of the asset class’s name (see Figure 16) to open the Upload screen (see Figure 17).

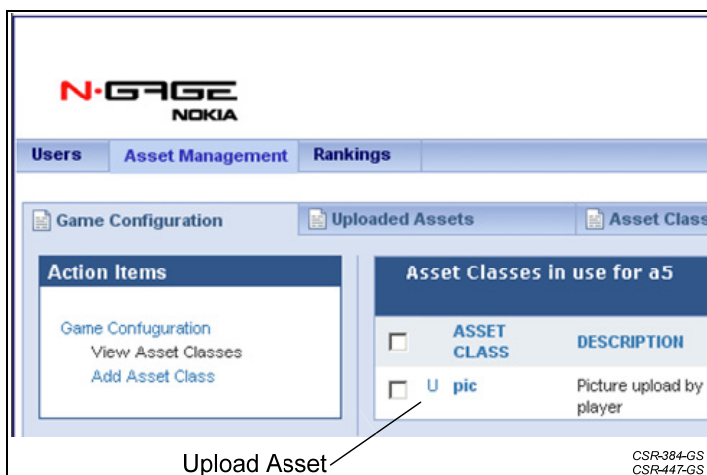


Figure 16: Link to Upload Asset screen.

- In the Upload screen, fill in the details for the asset to be uploaded into this asset class. See Figure 17 for an example followed by descriptions of the field and recommendations on how to use them.

Figure 17: Uploading an asset that contains metadata.

- Description:** Briefly describe the asset.
- Metadata:** Include metadata (optional). Metadata can be any string. To allow filtering of assets by metadata properties, you need to provide metadata in XML format. Example of proper XML syntax (see also “Metadata is XML?” under step 4 in Section 4.1, “Creating an asset class for a game.”):

```
<asset>
  <client-data>
    <myAvatar>myGreenAvatar</myAvatar>
  </client-data>
</asset>
```

**Note:** It is also possible to submit metadata that is not in XML format. However, to use filters when searching for assets through requests to the `listAssets(...)` method, it is necessary to write the asset metadata using the valid XML format. See the Asset Management section in the `client` package Javadocs of the SNAP Mobile SDK for uses of `AssetManager.listAssets(...)` and `AssetManager.uploadAsset(...)`. One example retrieves an asset (in this case, an avatar) with a metadata property “McQueen.” Note that the CSR tool does not validate the XML format, so it is important to check the syntax of the data in Metadata field before uploading an asset through the tool. It is also a good practice to validate the asset metadata syntax on the game client before uploading it.

- **Locale:** Type 10 for English. To find other locale IDs, see “Appendix: Locale identifiers.”
- **File:** Type the directory path or browse to the file to upload. For example, you might navigate to an image called `myAvatar.jpg`:



Figure 18: Sample asset called `myAvatar.jpg`.

5. Click the **Submit** button. A message with the `AssetID` appears if it was successfully uploaded (see Figure 19). Otherwise, you receive an error notification.

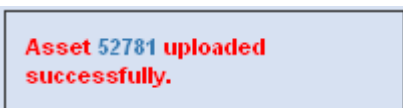


Figure 19: Upload notification.

6. View the new asset by following these steps:
  - a. Click the **Uploaded Assets** tab.
  - b. Search for the new asset by selecting your game and changing the validation status to **Passed** (or to the validation status that applies to the asset). See the new asset (`myMartian`) with the ID 52781 in Figure 20.

| Uploaded Assets |      |                   |       |                    |              |                   |               |                  |
|-----------------|------|-------------------|-------|--------------------|--------------|-------------------|---------------|------------------|
| GAME:           |      | ASSET CLASS :     |       | VALIDATION STATUS: |              | UPLOAD STATUS:    |               | VIEW RESULTS AS: |
| Any Game        |      | All asset classes |       | Passed             |              | Complete          |               | Text List        |
| ASSET ID        | GAME | DATE CREATED      | USER  | DESCRIPTION        | SIZE (BYTES) | VALIDATION STATUS | UPLOAD STATUS |                  |
| 52781           | 2    | 2008-05-23        | 12570 | myMartian          | 1584         | Passed            | C             |                  |

Figure 20: Viewing an uploaded asset.

The asset links to an **Edit Asset** screen through the asset ID link (52781 in Figure 20) and to a **User Details** screen through the User ID link (12570 in the figure). It also provides the game class ID (2 in the figure) for the game to which the asset is associated, the byte size of the asset, and other identifying data. Note that the upload status of “C” stands for complete. “U” stands for uploading.

**Note:** You can also view asset images and change the validation status of an asset from the asset search results by selecting to view results as a **Thumbnail list** (see Figure 21 and Figure 22).

| Uploaded Assets |  |                   |  |                    |  |                |  |                  |
|-----------------|--|-------------------|--|--------------------|--|----------------|--|------------------|
| GAME:           |  | ASSET CLASS :     |  | VALIDATION STATUS: |  | UPLOAD STATUS: |  | VIEW RESULTS AS: |
| Any Game        |  | All asset classes |  | Passed             |  | Complete       |  | Thumbnail List   |

Figure 21: Selecting to view results as a Thumbnail list.

| Uploaded Assets |                                                                                                      |      |      |              |       |             |              |                   |               |
|-----------------|------------------------------------------------------------------------------------------------------|------|------|--------------|-------|-------------|--------------|-------------------|---------------|
| ASSET ID        | VALIDATION STATUS                                                                                    | VIEW | GAME | DATE CREATED | USER  | DESCRIPTION | SIZE (BYTES) | VALIDATION STATUS | UPLOAD STATUS |
| 52781           | <input type="radio"/> Pending<br><input type="radio"/> Fail<br><input checked="" type="radio"/> Pass |      | 2    | 2008-05-23   | 12570 | myMartian   | 1584         | Passed            | C             |

Figure 22: Viewing the asset as a Thumbnail list.

#### 4.4 Managing uploaded assets

This section explains how to find, view, and change the validation status of an uploaded asset.

To manage uploaded assets:

1. From the main screen of the CSR tool, click the **Asset Management tab**.
2. Click the **Uploaded Assets tab** to open an asset search screen.
3. Search for the asset. You can use step 6.b in Section 4.3, “Uploading an asset into an asset class”, as a guide for completing this step. See Figure 20 for an example of a list of assets.
4. Click the link to the ID of the asset to open the Edit Asset screen. For example, you might click Asset ID 52768 in Figure 20.
5. View and manage the asset. For example, you can:
  - Modify the asset and click the Submit button that is located on the top-right portion of the Edit Asset screen to change the validation status of the asset. For example, if the asset is not longer supported, change its status to Failed.

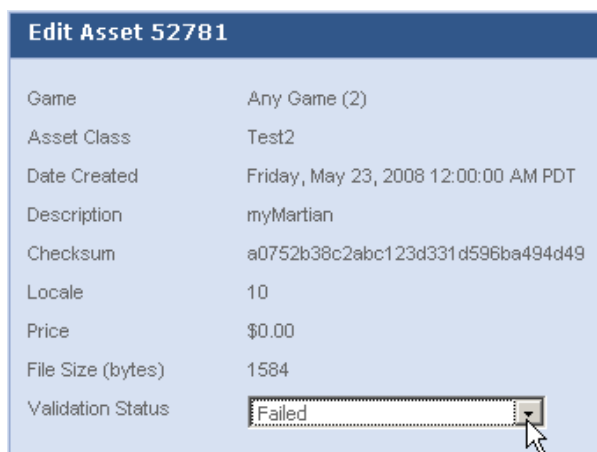


Figure 23: Changing the validation status of an asset to Failed.

**Note:** After making this modification to the asset, you can search for the updated asset using the new validation status. You can use step 6.b in Section 4.3, “Uploading an asset into an asset class”, as a guide to performing a search.

- Check the byte size and other properties of the asset. Note that Price is not supported. See Figure 23.
- View the metadata of an asset.

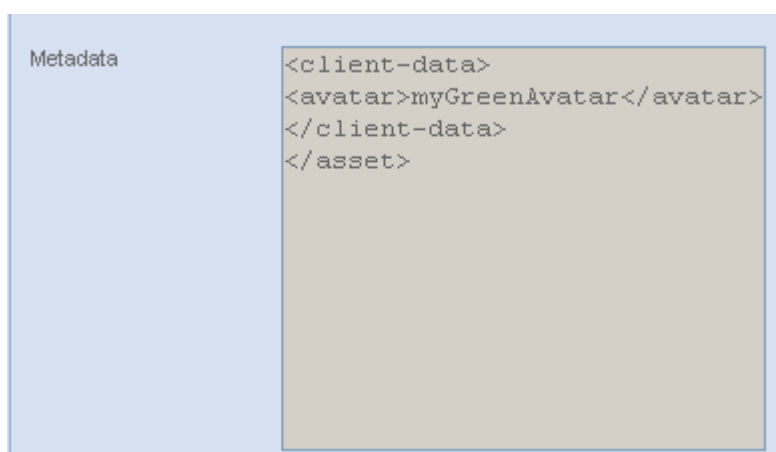


Figure 24: Viewing the metadata of an asset.

- Download the asset to view its content in a separate window. Note that this feature does not download the asset to the game.



Figure 25: Viewing the content of an asset.  
See Figure 18 for an example of asset content.

## 5 Managing rankings

This chapter provides information on how to configure rankings for a game. This functionality supports the ranking retrieval and score reporting methods described in the Introduction.

To open to the Rankings portion of the CSR tool:

1. From the main screen of the CSR tool, click the **Rankings** tab. Figure 26 provides an illustration of the Rankings tab content.



Figure 26: Rankings tool

2. Use the features in the Rankings tab to specify and configure a Highscore board and/or a Highscore Filter board for your game:
  - **Highscore board (highscores scoring system):** Requires the specification of stats for your game. This board allows a game to submit statistics used to process player rankings, such as the score at the end of the game, the total number of wins by a particular player, and other data relevant to ranking players.
  - **Highscore Filter board (hsfilter scoring system):** Requires the specification of stats and filters for your game. This board allows the game to submit filtered statistics, for example, to submit scores for a golf game on one of many golf courses (where, for example, `golf_course` is the filter, and the value represents a particular course, such as the Palm Springs course or a course located in Ireland or in Hawaii). Such a submission allows for the ranking of a players based on which golf course they used when playing the game.

**Note:** SNAP Mobile does not support other scoring systems that are visible in the CSR tool: `playerskills`, `bank`, `events`, `ngps`, or `ngpsglobal`. Do not attempt to use them.

### 5.1 Setting up a Highscore board for a game

This section explains how to use the CSR tool to configure stats for a Highscore board. See the procedure in “Managing rankings” for a brief description of this board. See also the “Rankings” section of the `client` package documentation in the SNAP Mobile SDK Javadocs for detailed information on ranking boards.

To specify stats for your game in the Highscore board:

1. After opening the Ranking tab, click your game to open the list of scoring systems. See Figure 26 for an example.

- Click the Highscore board link (`highscores`) to open the configuration screen for this board. See Figure 27.

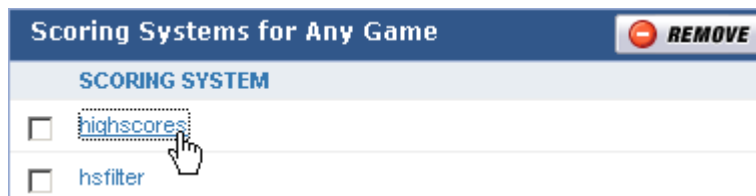


Figure 27: Selecting the Highscore board.

- Complete the form to specify one or more stats for your game. See Figure 28 for an example.

| ID | STAT NAME | ORDERING     | STAT TYPE         | LIMIT MIN | LIMIT MAX | STAT DESCRIPTION |
|----|-----------|--------------|-------------------|-----------|-----------|------------------|
| 1  | SCORE     | High is Best | Replace if Better |           | 1000      | SCORE            |
| 2  | WINS      | High is Best | Accumulate        |           |           | WINS             |

Figure 28: Specifying stats for a game.

In this example, there are two stats: SCORE and WINS:

- SCORE:** Assuming the game is a golf game, this stat is set to an order of Low is Best and stat type of Replace if Better so that lower scores will replace higher scores that the player submits to the board. For example, if a player submits a score of 68 after scoring 79 in a previous game, the score of 68 will overwrite the 79. If the player submits a score of 75 after submitting a score of 68, the new score will not overwrite the score of 68. The SCORE stat also prevents the submission of a very high score (1000), so the board will accept a score of 999 but not a score of 1000 or more.
- WINS:** This stat is configured to provide the total number of wins by a player (stat type is Accumulate) and to base rankings on the highest number of wins (ordering is High is Best).

The following list describes the fields in Figure 28:

- ID:** Identifies the stat numerically.
- Stat Name:** Provides the name of the stat.
- Ordering:** Specifies a ranking order for the values submitted for the stat.
  - High-is-best** replaces the existing value for a given stat/value pair with a new value if the new value is higher than the existing value; for example, as in a basketball or football game.
  - Low-is-best** replaces the existing value for a given stat/value pair with a new value if the new value is lower than the existing value; for example, as in a golf game.
- Stat Type:** Provides a rule for modifying the values of stat/value pairs when the game submits them.
  - Replace:** Overwrites the existing value with a new value. This option checks the following rules before replacing the existing value: Ranking Order, Minimum Limit, Maximum Limit.
  - Accumulate:** Adds (if the value is a positive number) or subtracts (if the value is a negative number) the new value from the existing value. For this stat type, the maximum (see Limit Max,) applies to the sum of all values that the user has submitted. For example, if the maximum is twenty (20), and a user submits scores of ten (10) in one game and five (5) in the next game, the total is fifteen (15), so the submissions do not exceed the maximum. However, if the user plays another game and submits a third score of six (6) for a total of 21, then the third submission exceeds the maximum, so the ranking system does not record the third score. As a

consequence, games are unlikely to use a maximum limit when accumulating scores, wins, or other ranking submissions.

- **Overwrite:** Overwrites the existing value with the new value without checking the Ranking Order or entering a Minimum Value or a Maximum Value.

**Note:** For the stat types `Replace` and `Overwrite`, the maximum limit applies per ranking submission. For example, if the maximum is twenty (20), and the ranking configuration replaces or overwrites the submitted scores, then a user can submit scores of ten (10) in one game and fifteen (15) in the next game. However, if the user attempts to submit a score of 21, the ranking system will not record this excessive score.

- **Limit Min:** Identifies the minimum allowable value for this stat/value pair per ranking submission. (Optional)
- **Limit Max:** Identifies the maximum allowable value for this stat/value pair (Optional). Note that the maximum applies per submission if the stat type is `Replace`. However, it applies to the accumulated total if the stat type is `Accumulate`, which can lead to undesirable consequences unless used properly. Read the description of `Accumulate` for additional information on how a value in this field is used to limit accumulations.
- **Stat Description:** Provides a brief description of the stat, often the name of the stat as it appears in the game or on a Web site.

4. Submit the stats.

You can now use the ranking methods in the SNAP Mobile API to submit stats and retrieve player rankings from the Highscore Board for your game. See “Introduction” for a list of these methods.

**Note:** New ranking stat changes will be available for use approximately 2 hours after you input them.

## 5.2 Setting up a High Score Filter board for a game

This section explains how to use the CSR tool to configure stats for a Highscore Filter board. See the procedure in Chapter 5, “Managing rankings”, for a brief description of this board. See also the “Rankings” section of the `client` package documentation in the SNAP Mobile SDK Javadocs for detailed information on ranking boards.

To specify filter data on the Highscore Filter Board for your game:

1. After opening the Ranking tab, click your game to open the list of scoring systems. See Figure 26 for an example.
2. Click the Highscore Filter board link (`hsfilter`) to open the configuration screen for this board. See Figure 27 for an example that includes the `hsfilter` link.
3. Complete the filter form (titled “Enter Filter stat”) to specify one or more filters for your game. See Figure 29 for an example.



Figure 29: Specifying filters for a game

4. Specify stats for your Highscore Filter Board.

The Highscore Filter Board requires a list of the stats that you plan to filter. You can use step 3 in Section 5.1, “Setting up a Highscore board for a game”, as a guide to completing this data because the interface is the same. Note that you must configure both stats and filters to use this board.

The game submits filtered scores through a request to the `reportScores(...)` method. Filters take a string value. Note that developers must keep track of the values for each filter, for example,

whether the golf courses are understandable names (for example, Hawaii, Ireland, California), or string integers (such as 1 for Hawaii, 2 for Ireland, and 3 for California).

5. Submit the filters and stats.

You can now use the ranking methods in the SNAP Mobile API to submit stats and retrieve player rankings from the Highscore Filter Board for your game. See “Introduction” for a list of these methods.

**Note:** New ranking stat and filter changes are available for use approximately 2 hours after you input them.

## Appendix: Locale identifiers

Locale identifiers can either be Symbian values corresponding to the `TLanguage` enumeration, or they can be specified as defined in RFC3066, which is a string of the format `<language>-<country>`. For US English, this is `en-US`, which is the default.

Table 1 lists the Symbian locale IDs along with their descriptions.

**Note:** These codes are also available at <http://www.forum.nokia.com>.

Table 1: Symbian locale identifiers.

| ID | Description                         |
|----|-------------------------------------|
| 00 | Test                                |
| 01 | English: United Kingdom             |
| 02 | French                              |
| 03 | German                              |
| 04 | Spanish                             |
| 05 | Italian                             |
| 06 | Swedish                             |
| 07 | Danish                              |
| 08 | Norwegian                           |
| 09 | Finnish                             |
| 10 | English: United States              |
| 11 | French: Swiss                       |
| 12 | German: Swiss                       |
| 13 | Portuguese                          |
| 14 | Turkish                             |
| 15 | Icelandic                           |
| 16 | Russian                             |
| 17 | Hungarian                           |
| 18 | Dutch                               |
| 19 | Belgian: Flemish                    |
| 20 | Australian                          |
| 21 | Belgian: French                     |
| 22 | Austrian                            |
| 23 | NewZealand                          |
| 24 | French: International               |
| 25 | Czech                               |
| 26 | Slovak                              |
| 27 | Polish                              |
| 28 | Slovenian                           |
| 29 | Chinese: Taiwan                     |
| 30 | Chinese: Hong Kong                  |
| 31 | Chinese: People's Republic of China |
| 32 | Japanese                            |
| 33 | Thai                                |
| 34 | Afrikaans                           |
| 35 | Albanian                            |

| ID | Description            |
|----|------------------------|
| 36 | Amharic                |
| 37 | Arabic                 |
| 38 | Armenian               |
| 39 | Azerbaijani            |
| 39 | Tagalog (Azerbaijani)  |
| 40 | Belarussian            |
| 41 | Bengali                |
| 42 | Bulgarian              |
| 43 | Burmese                |
| 44 | Catalan                |
| 45 | Croatian               |
| 46 | English: Canadian      |
| 47 | English: International |
| 48 | English: South African |
| 49 | Estonian               |
| 50 | Farsi                  |
| 51 | French: Canadian       |
| 52 | Scots Gaelic           |
| 53 | Georgian               |
| 54 | Greek                  |
| 55 | Greek: Cyprus          |
| 56 | Gujarati               |
| 57 | Hebrew                 |
| 58 | Hindi                  |
| 59 | Indonesian             |
| 60 | Irish                  |
| 61 | Swiss Italian          |
| 62 | Kannada                |
| 63 | Kazakh                 |
| 64 | Khmer                  |
| 65 | Korean                 |
| 66 | Lao                    |
| 67 | Latvian                |
| 68 | Lithuanian             |
| 69 | Macedonian             |
| 70 | Malay                  |
| 71 | Malayalam              |
| 72 | Marathi                |
| 73 | Moldavian              |
| 74 | Mongolian              |
| 75 | Norwegian Nynorsk      |
| 76 | Brazilian Portuguese   |
| 77 | Punjabi                |
| 78 | Romanian               |
| 79 | Serbian                |
| 80 | Sinhalese              |
| 81 | Somali                 |
| 82 | Spanish: International |

| ID  | Description                               |
|-----|-------------------------------------------|
| 83  | Spanish: Latin American                   |
| 84  | Swahili                                   |
| 85  | Swedish: Finland                          |
| 86  | Tajik                                     |
| 87  | Tamil                                     |
| 88  | Telugu                                    |
| 89  | Tibetan                                   |
| 90  | Tigrinya                                  |
| 91  | Turkish: Cyprus                           |
| 92  | Turkmen                                   |
| 93  | Ukrainian                                 |
| 94  | Urdu                                      |
| 95  | Uzbek                                     |
| 96  | Vietnamese                                |
| 97  | Welsh                                     |
| 98  | Zulu                                      |
| 99  | Other                                     |
| 157 | English (APAC Taiwan)                     |
| 158 | English (APAC People's Republic of China) |
| 160 | English (APAC Hong Kong)                  |
| 161 | English (APAC Japan)                      |

## 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](#).