Mobile Client to GNSDK for Mobile Migration Guide

Version 1.1

Published: 6/5/2014 6:22 PM

Gracenote, Inc.

2000 Powell Street, Suite 1500

Emeryville, California

94608-1804

http://www.gracenote.com

Table of Contents

Introduction

This document describes important differences between Gracenote Mobile Client 3.x and GNSDK for Mobile 1.x. Follow these guidelines when migrating applications.

Note

This is an early draft of this document. Please contact Gracenote Global Support & Services for additional information.

Object Oriented Query Objects

Mobile Client and GNSDK for Mobile arrange objects for performing Gracenote queries and operations differently. Mobile Client provided a single class that contains methods for each individual query or operation type, GNSDK for Mobile provides individual classes for each query or operation.

The object-oriented approach also allows different queries or operations to execute and complete independently, meaning there is no central queuing mechanism within GNSDK for Mobile, allowing your application greater control over the execution of Gracenote functionality.

Regionalization

Mobile Client provided global regionalization configuration settings such as "content.lang", "content.country" and "content.era.level".

GNSDK for Mobile provides access to Gracenote's full locale management interface. To configure GNSDK for Mobile to deliver regional data your application must:

- Load one or more locales with appropriate region and language settings, typically derived from user preferences or device settings
- Associate a locale with responses received from GNSDK for Mobile, by setting a default locale or setting the locale of each response manually
- Updating loaded locales, fetch updated of locale data from Gracenote
- Access the full suite of regions and languages Gracenote supports in all levels of granularity (for example detailed and simplified genres)
- Access all of Gracenote's Lists for building advanced application features (such as a language selection screen where each language is expressed in native language and script)
- Determine when locale data is downloaded from Gracenote service, or if locale data is delivered with the application when deployed

Locale Updates

Gracenote is constantly updating music classifications to reflect changes world wide. To keep your application current it will need to update it's locale data from time to time. GNSDK for Mobile can detect when an application should update a locale and call an IGnSystemEvents delegate method providing the GnLocale object. Your application can register for system events via a GnManager instance.

Lists

GNSDK for Mobile also provides access to Gracenote Lists, which provides more granular access to Gracenote data. Typical applications are adequately served by the Locale interface and need not access Lists directly.

How should my application implement regionalization?

Because every application is different GNSDK for Mobile provides many options for implementing regionalization. Typical applications can follow the implementation provided in the sample application, but you should work with your Gracenote support representative to ensure your application implements the most appropriate solution.

Recognizing Audio from a Microphone

Mobile Client provides IDNow functionality which continuously listens to an audio stream readying the audio for identification when the application requires it. GNSDK for Mobile provides the same functionality via GnMusicIdStream object

GnMusicIdStream can receive raw audio provided manually, where the application routes audio from the stream source to GnMusicIdStream; it can also or automatically pull audio from an audio source that implements the IGnAudioSource interface. On some platforms Gracenote provides a GnMic class that implements IGnAudioSource, providing a convenient way for identifying music in the user's environment. Developers are free to provide their own implementations of this interface connecting GnMusicIdStream with any type of audio source.

- Create your own microphone recording code that provides the raw audio to GnMusicIdStream manually, analyzing the raw audio before doing so
- Create a fixture class that sits between GnMic and GnMusicIdStream which is a pass-through imlementation of IGnAudioSource. The fixture can intercept raw audio as it passes.

Recognizing Audio Files

Mobile Client provides AlbumID functionality for identifying audio files present on the device. The application can identify a single audio file, all audio files in a directory and sub-directories, or all audio files specified by file information objects.

GNSDK for Mobile provides AlbumID functionality via GnMusicIdFile object. The application adds audio files to GnMusicIdFile object giving it a unique identifier and providing, where available, metadata and a fingerprint.

With all files added to GnMusicIdFile object they can be identified using Gracenote's file identification algorithms known as Track ID, Album ID or Library ID. Library ID is the closest to the AlbumID functionality presented in Mobile Client.

Canceling Operations

Mobile Client provides ability to cancel a single operation or cancel all operations.

GNSDK for Mobile provides ways to cancel objects that could make a query to Gracenote Service. Query objects typically have a cancel method on the object interface but also provide a "canceller" object to delegate methods. This allows applications flexibility and convenience in how cancel is implemented.

Due to fundamental differences in architecture GNSDK for Mobile cannot offer "cancel all" functionality. The concurrent nature of GNSDK for Mobile allows different query objects to operate independently, meaning there is no central queuing mechanism for ongoing query objects within the SDK, without such as mechanism the SDK cannot provide a "cancel all" method.