![Gracenote logo](gracenote.)

GNSDK for Mobile Android Release Notes

Version 1.1.9

Published: 12/28/2015 3:38 PM

Table of Contents

Overview

This document describes the Gracenote GNSDK for Mobile Software Development Kit for the Android platform. It lists the delivered package contents, highlights, and release limitations. For more information, see the GNSDK for Mobile Android Documentation.

Contents of SDK

The release is distributed as a compressed archive file, gnsdk-mobile-android-1.1.x.x.zip, where x.x is the current release number as listed on the title page of this document. The zip file contains the following components:

- GNSDK for Mobile SDK

- Sample application

- Documentation

    o GNSDK for Mobile Help System (HTML5), including, Getting Started, Implementing Applications, and API Reference content.

    o GNSDK for Mobile Developers Guide (PDF): Printable version of the Help system without the API Reference.

    o Mobile Client to GNSDK for Mobile Migration Guide (PDF)

    o Legal Notices (PDF)

Products Included

GNSDK For Mobile includes the following products:

- Gracenote MusicID (CD TOC, Text)

- Gracenote MusicID-File (Audio-file-based recognition)

- Gracenote MusicID-Stream (Audio-stream recognition)

- Gracenote Playlist (Automated Playlist generation and MoodGrid support)

Requirements

- Android devices capable of running architectures armeabi, armeabi-v7a, arm64-v8a, or x86

- SDK supports Android Platform Version 2.3.3 (API Level 10) or higher

- Sample application requires Android Platform Version 2.3.3 (API Level 10) or higher

- GNSDK for Mobile SDK libraries must be included within each application (`.apk`)

- Google APIs by Google Inc. (Android API Level 10) must be installed to compile and run sample application.

- Locale Space requirements

    o   < 2Mb for Music only locale

    o   < 6Mb for Music and Playlist locale

- Local Bundle space requirements

    o   Optionally, approximately 21 MB of memory is needed for the local bundle storage

- It is strongly recommended that only bundles numbered 1829.b and greater be used with this release

Changes In This Release

- Numerous performance improvements in Playlist and Moodgrid. Improvement is seen in collection load times and moodgrid heat map generation. Playlist collection summaries created with this version of GNSDK for Mobile 1.1 will not work with older versions.

- Fixed potential crash after Playlist locale loaded in list related operations such as retrieving genre, origin, artist type, era, mood or tempo data from result object.

Limitations

- If you are using C++ or Java in your Android development, you must link the libgabi++_shared.so library into your application. This library is available in the Android NDK beginning with version 8d.

- Content Delivery

    o Genre Art is not currently supported

    o GnResponse objects rangeTotal, rangeStart, and rangeEnd return 0 when a track with no cover art is returned

- Playlist of zero tracks is generated when "more like this" is requested

- "An error occurred in the communications subsystem" message returned when not connected to internet

- Setting a log size limit using gnLog.options(new GnLogOptions().maxSize()) for GnLog archive does not work and instead of creating an archived log file on reaching the maxSize the log file continues to grow.

- Concurrent ingestion of bundles is not supported and can lead to corrupt local MusicID-Stream databases resulting in error "Invalid Data" being returned when attempting a local lookup.

- GnMusicIdStream lookup mode cannot be changed after audioProcessStart is invoked. Applications must set the lookup mode immediately after instantiating a GnMusicIdStream object and will not be able to change it. If lookup mode must be changed the application can destroy its GnMusicIdStrdeam instance and create another.

- GnMusicIdStream setting resultSingle may return more than a single match for GnMusicIdStream identify operations.

- SDK will perform an IP resolve procedure on any network query that follows a MusicID-Stream query that was cancelled while accessing the network.

Release History

Version 1.0

- Includes support for Gracenote MoodGrid and Playlist products
- Improved language support for MusicID-Stream queries
  - Language and script preferences surfaced to developers
- Support for multiple Client IDs
- Requires GNSDK License File (please contact your Gracenote Engagement Manager)
- Decoding of MP3, AAC, 3GP (AAC) file formats for MusicID-File only supported for Android Platform Version 4.1 (API Level 16) and higher

Version 1.0.1

- Support for Intel Atom processor (x86)

Version 1.0.2

- Added default/invalid value to GnLocaleGroup enum being kLocaleGroupInvalid
- Added GnPlaylistCollection method public GnString statementAnalyzeIdent (String pdlStatement, String mediaIdentifier) as an Experimental API to be used to debug PDL statements by applying a PDL statement to a specific media item within a playlist collection summary.
- Added GnDataObject method matchDuration
- Removed Invalid constructor GnList.GnList()
- Changed enumeration value from kGnMusicIdFileInfoStatusError to kMusicIdFileInfoStatusError as naming convention differed for GnMusicIdFileInfoStatus values
- Changed enumeration value from kGnMusicIdFileInfoStatusResultAll to kMusicIdFileInfoStatusResultAll as naming convention differed for GnMusicIdFileInfoStatus values

Version 1.1.0

- A Sample Android widget is included (together with a Sample application)
- GnMusicIdStream
- `audioProcessStop()` marked as experimental. Its behavior has been modified to stop audio processing and pending identifications. In near future this may change.
-

  Cancel implementation overhauled resulting in changes in the API from
  ```
  o  public void identifyfCancel()
  o  public boolean isIdentifyCancelled()
  ```
  to
  ```
  o  public void identifyCancel() throws
     com.gracenote.gnsdk.GnException
  ```

- Bundle Ingestion

  Bundle ingestion APIs have changed to facilitate:
  - Streaming ingestion of bundles meaning bundles do not have to be entirely downloaded before ingestion can be started
  - Concurrent ingestion of multiple bundles

  The following ingestion related APIs have been removed:
  - `GnBundle`
  - `IGnBundleIngestStatusEvents`
  - `IGnBundleSource`
  - `GnLookupLocalStream` methods:
    - `public long storageCount() throws com.gracenote.gnsdk.GnException`
    - `public void ingest( IGnBundleSource ingestSource ) throws com.gracenote.gnsdk.GnException`

  The following ingestion related APIs have been added:
  - `GnLookupLocalStreamIngest`, used to ingest a fingerprint bundle
  - `IGnLookupLocalStreamIngestEvents`, bundle ingestion events delegate

- Logging APIs Updated

  Logging APIs have been updated and are no longer experimental. See sample and documentation for more information.

- GnLocale

  The following locale information access methods have been removed:
  - `public GnLocaleGroup group() throws com.gracenote.gnsdk.GnException`
  - `public GnDescriptor descriptor() throws com.gracenote.gnsdk.GnException`
  - `public GnLanguage language() throws com.gracenote.gnsdk.GnException`
  - `public GnRegion region() throws com.gracenote.gnsdk.GnException`

  and replaced by a new API,
  - `public GnLocaleInfo localeInformation()`

  The API to determine which locales are available from locale storage has been changed from,
  - `public static GnLocaleInfoIterable localesAvailable()`

  to,
  - `public static GnLocaleInfoIterable storedLocales()`

- GnMusicIdFile

  Cancel method has been changed from,
    o `void setCancel(boolean bCancel)`

  to,
    o `void cancel()`

  There is no need for an application to reset the internal cancel flag, so it does not need a boolean.

  Method `isCancelled()` is removed as there is no meaning keeping cancel state for an object that lives beyond a single identification query.

- GnAlbum

  Remove access to script display string. This information is not yet fully supported by the SDK.
    o `public String scriptDisplay()`

- GnTrack

  Removed methods for obtaining the corresponding MusicID-File matched identifier and matched filename,
    o `public String matchedIdent()`
    o `public String matchedFilename ()`

  Instead a single GnTrack object can match to multiple MusicID-File file info objects, so a new API that returns an iterator for all matched file info identifiers is provided,
    o `public GnStringValueIterable matchedIdents()`

- List Element and List Element Child Iterator Access

  Methods to get list element iterator no longer throw GnException. Changed from,
    o `public GnListElementIterable listElements(long level)`
       `throws com.gracenote.gnsdk.GnException`
    o `public GnListElementChildIterable children() throws`
       `com.gracenote.gnsdk.GnException`

  to,
    o `public GnListElementIterable listElements(long level)`

    o `public GnListElementChildIterable children()`
- GnAssetIterable
  Method to get iterator has been made consistent with other iterable. Method changed from,
       `public GnAssetIterator begin()`
  to
       `public GnAssetIterator getIterator()`

Version 1.1.1

- Provides callbacks for audioProcessStart (which indicates when GnMusicIdStream is ready for an identify call) and audioProcessStop.

Version 1.1.2

- Improved performance of Cancel operation when SDK is performing a network operation

- GnMusicIdStream deprecated constructor removed. Applications will now need to use a GnMusicIdStream constructor that takes an audio preset denoting the audio is from a microphone or from a radio signal.
  ```
  public GnMusicIdStream( GnUser user, IGnMusicIdStreamEvents
  pEventDelegate ) throws com.gracenote.gnsdk.GnException
  ```

- GnMusicIdStream cancel method is blocking and does not return until the pending identification is cancelled.

- Fixed crash defect when near silent audio is fed to GnMusicIdStream identify.

Version 1.1.3

- Fixed defect in PDL which caused an "Unexpected call to function" exception

- Fixed crash in `hasNext()` for many iterator types (including `GnPlaylistCollectionIdentIterator.hasNext()`)

Version 1.1.4

- Fixed defect in SDK where cancel request was being ignored in certain cases

- Modified sample app Library ID functionality such that if cancel is pressed during user collection analysis the operation ceases and does not proceed to call GnMusicIDFile.doLibraryId

- Fixed crash seen with Android decoder on Android 4.1 when files of unsupported media type are passed to SDK

- Fixed issue in SDK where under certain conditions the GnMusicIdStream identify operation would return the previous track matched

Version 1.1.5

- Rolled back modifications to language preferences for online Text and file based fingerprint queries. The modifications were causing issues in some regions in which the preferred result was being filtered out. This did not impact MIDS fingerprint queries and the language preference improvements are still in place for MIDS.

Version 1.1.6

- GnAudioFile improved to protect against Android native decoder crashes caused by some audio files (such as ALAC files) occurring on some devices

- Eliminated symbol clashes with SQLite. This was causing symbol clashes in some cases where developers were using another instance of SQLite in their application

- Fixed Playlist PDL LIMIT clause to skip missing attributes

- Fixed Playlist collection summaries generated so that they are compatible with previous version of SDK

Version 1.1.7

- Fixed Java Security Vulnerability (CVE-2015-2004) in the Java layer of the GNSDK which is potentially exploitable if your application is serializes GnException and stores or transmits the serialized object.

- Java marshaling shared library name changed to include GNSDK for Mobile product version number.

Version 1.1.8

- Certified for Android 6.0 (Marshmallow)

- Sample application updated to use HttpURLConnection

- Android architecture arm64-v8a added; therefore libraries built with NDK 10e (previous release use 8e)

- Various internal updates, fixes and optimizations

Confidentiality Notice

This document is confidential information of Gracenote, Inc., and is for Gracenote employees and intended recipients only. Any dissemination, distribution, or copying of this communication is strictly prohibited.

Gracenote®, the Gracenote logo and logotype, and the "Powered by Gracenote" logo are either registered trademarks or trademarks of Gracenote, Inc., in the United States and/or other countries.