NexEditorSDK for Android

Technical Reference Manual

Version 1.7.0

Generated November 23, 2016

NEXSTREAMING

Contents

Chapter 1

NexEditor™SDK for Android

1.1 Legal Notices

Disclaimer for Intellectual Property

This product is designed for general purpose, and accordingly the customer is responsible for all or any of intellectual property licenses required for actual application. NexStreaming Corp. does not provide any indemnification for any intellectual properties owned by third party.

Copyright

Copyright for all documents, drawings and programs related with this specification are owned by NexStreaming Corp. All or any part of the specification shall not be reproduced nor distributed without prior written approval by NexStreaming Corp. Content and configuration of all or any part of the specification shall not be modified nor distributed without prior written approval by NexStreaming Corp.

1.2 Abstract

The NexEditor™ SDK provides video editing services, including support for images and text, with various theme, text, and transition effects, that allow application developers to build custom applications with video editing features efficiently. The NexEditor™ SDK has been built to be reliable and robust without any sacrifice in performance, and has proven compatibility with international standards.

This documentation is a work in progress and details will continually be added to provide additional information to developers.

Note: NexEditor™ SDK APIs were developed based on JAVA.

1.3 NexEditor™ SDK Capabilities and Limitations

Operating Systems:

- The NexEditor™ SDK supports Android 4.4 (API Level 19) and above.

Input File Containers:

- Video : MP4, 3GP and MOV
- Audio : MP3 and AAC

Input Image Format:

- PNG and JPEG

Input Video Codec:

- H.264

Input Audio Codec:

- AAC and MP3

Output File Container:

- MP4

Output Video Codec:

- H.264

Output Audio Codec:

- AAC

1.4 NexEditor™ SDK Concept

1.4.1 NexEditor™ SDK Main Features

Multi-Video Editing

- Photo : Make a video using a number of images with different formats.
- Video : Edit several videos and images in different formats at the same time.

Basic Editing

Most video editing can be done with the basic editing features, including the following: adjusting the duration of images, moving, zooming, panning, rotating images, extracting clips of video, and merging several videos into a single video file. Just by using these basic editing features, detailed edits to the millisecond are possible with time-adjusting APIs.

Instant Preview

Instant preview is possible because there is no need to convert the videos.

Export

The videos can be converted into various resolutions (1080p, 720p, 640p, 480p).

NOTE: Depending on the chipset, supported resolutions may vary.

Image Stickers on Video

Images can be overlaid on top of the videos or other images.

Various Themes

Various themes are provided to allow videos to be edited with effects set automatically.

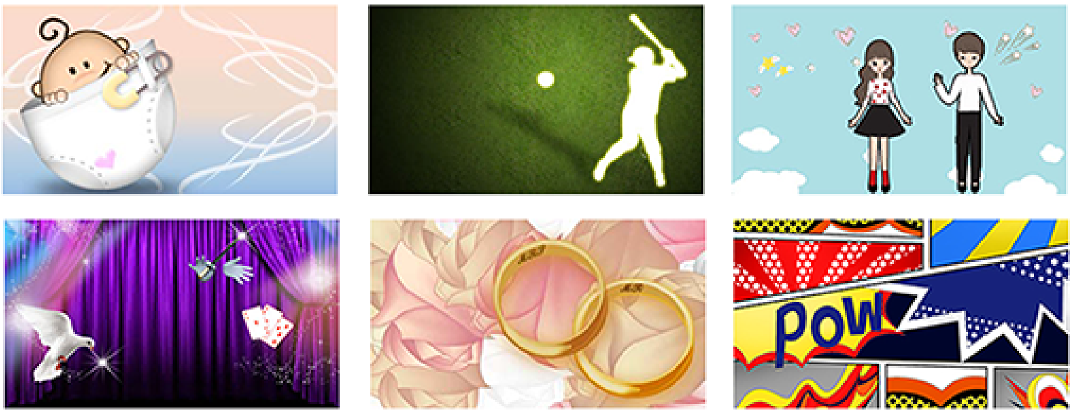


Figure 1.1: Theme Thumbnails

Various Transition Effects

Various transition effects including PIP transition, 3D transition, and Mask transition are provided.

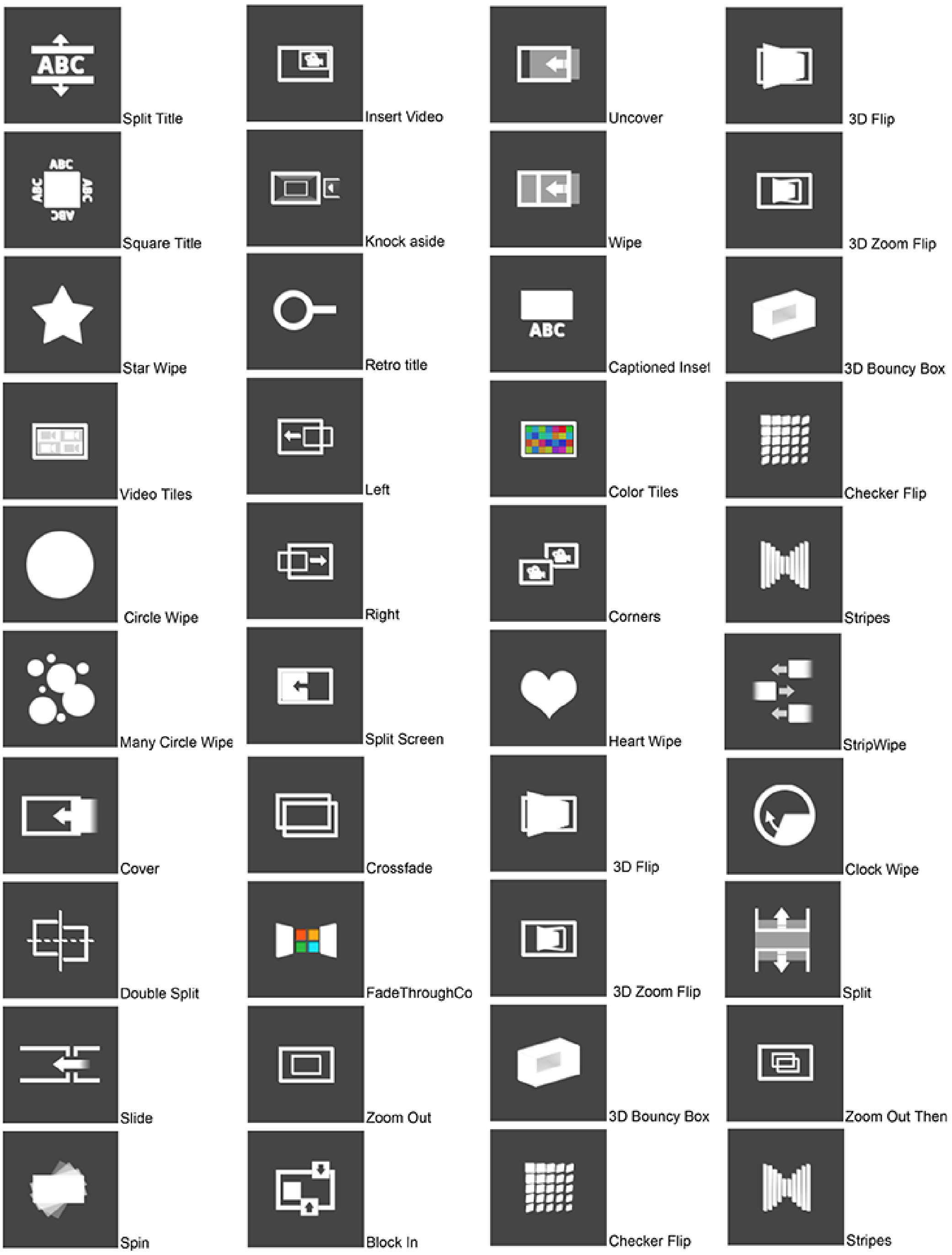


Figure 1.2: Transition Effect Icons

Various Clip Effects

Various clip effects and text effects for video and image clips are supported.

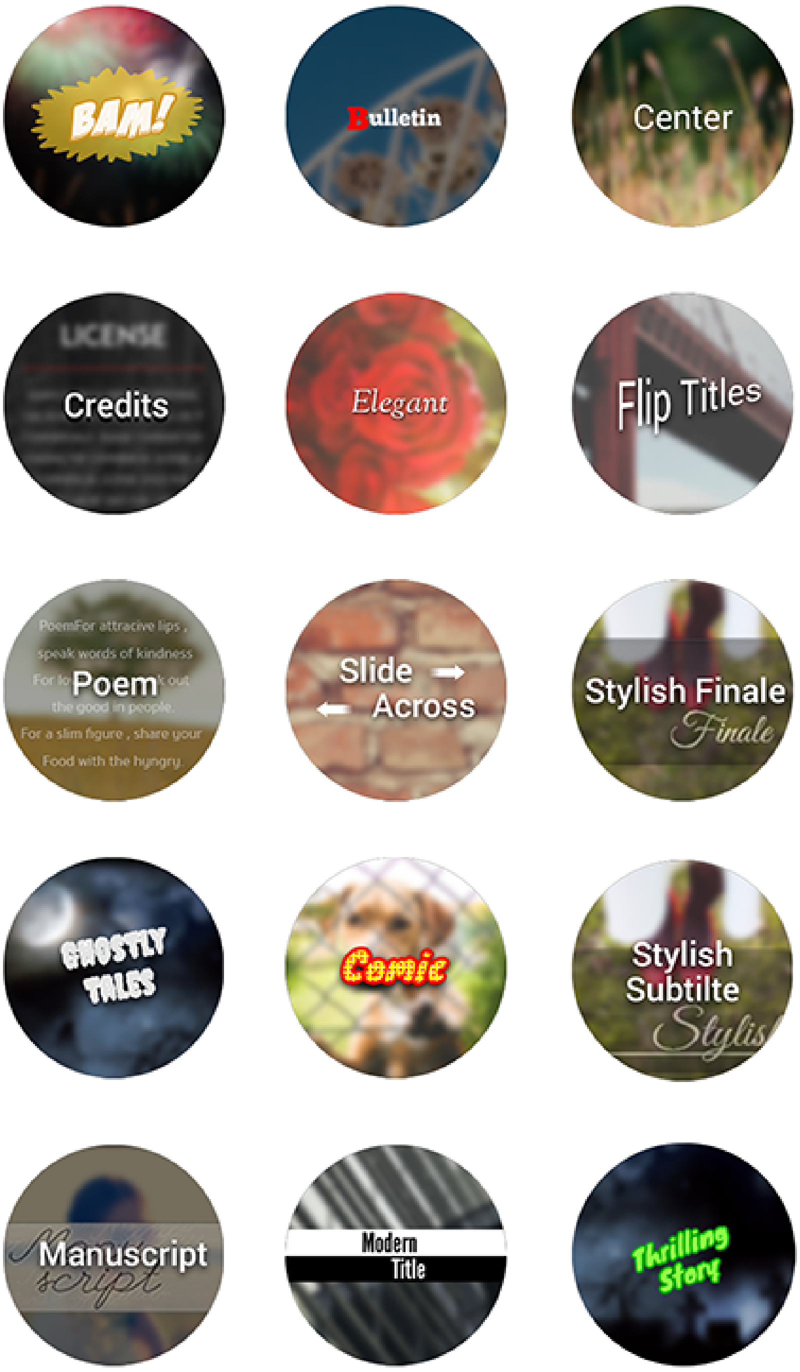


Figure 1.3: Clip Effect Icons

Speed Control

Speed control for video is supported. The speed of an extracted video clip can be adjusted from 0.25 quarter speed) to 2 times the speed of the original video. The audio pitch will not change even when the speed of a video is changed.

Color Adjustment

The color of video clips can be adjusted, including allowing fine tuning of:

- Brightness: Adjustment of video brightness,
- Contrast: Adjustment of video contrast,
- Saturation: Adjustment of video color saturation, and
- Tint: Adjustment of video tint (or use of color filters).

Multi-Audio Editing

Multiple audio files can be used when editing video.

Multi-Audio Track

Multiple audio files, to a maximum of three audio tracks, can be edited at the same time.

Audio Fade-in & Fade-out

At the beginning and end of a video clip, fade-in and fade-out effects can be applied to prevent audio from starting or ending awkwardly.

Audio Envelope

The volume of the audio can be adjusted to different levels in different sections of the content.

Video Thumbnail & Audio PCM Graph

- Video: Extracts thumbnail of the video clip.
- Audio: Extracts PCM Graph (Thumbnail) of the audio clip.

1.4.2 NexEditor™ SDK Terms

Clip

A clip can be an image clip (photos and pictures), a video clip (MP4), or an audio clip (MP3, AAC). Supported Formats:

- Image: File (media) formats (supported by the / that can be decoded by) Android framework.
- Video: MP4 (with H.264 for video and AAC for audio).
- Audio: MP3 and AAC.

Project

Collection of clips to be used and edited in the NexEditor™ SDK. Image clips and video clips are handled in the primary track and audio files are edited in a secondary track.

Clip Effect

Effect that can be added to a clip. Only one clip effect can be assigned to a clip at a time and the effect may or may not include text.

Transition Effect

An effect used to transition between one clip and a second clip. Depending on the transition effect added, clips may overlap during the transition.

Theme

Automatic application of simple effects to clips based around a theme. The clips of a project will be automatically edited to include the effects and transitions bundled in the theme. A theme is a package of transition effects, clip effects with text, and clip effects without text (that are all used to craft a story from the video clips to be edited). A basic theme is automatically applied when a project is created with the NexEditor&t SDK which later can be changed to other themes at any time. Specific theme effects are dependent on the theme so if it is changed for a project, the included theme effects will also change to the effects bundled with the newly selected theme.

Non-Theme Clip Effects

Clip effects that are not bundled in the themes. These independent clip effects can be used to replace the clip effects that are automatically set on a clip based around a theme. Therefore even the theme is changed, the independent effects will remain on the specific clip.

Non-Theme Transition Effects

Transition Effects that are not bundled in the themes. These independent transition effects can be used to replace the transition effects that are automatically set between the clips based around a theme. Even if the theme is changed, the independent transition effect will remain between the clips.

Engine

The main module, developed based on singleton pattern design, that's needed to preview and export a project.

Engine View

Android surface needed for previewing the engine. In the the NexEditor™ SDK, this has to be generated as a view.

Effect Library

Manages clip effects, transition effects, and theme effects. This module must be initialized when the program begins. The effect library operates asynchronously so it must be used after loading has completed. Developed based on singleton pattern design.

Effect Preview View

A view to preview any clip effect, transition effect, or theme effect without using the NexEditor™ SDK engine.

Effect ID

String IDs to identify effects and themes. NexEditor™ SDK IDs are stored in the Effect Library.

Speed Control

A function to control the playback speed of a video clip.

Trim

A function to select and cut a specific section of a video clip (instead of using the whole original clip).

Image Duration

The duration of time to play an image clip in a project.

Units

- Time: Unless otherwise indicated, all NexEditor™ SDK API time units are measured in `milliseconds`.
- Playback speed: The playback speed of a clip is controlled as a percentage. Normal speed will be 100% and doubled speed will be 200%. Input values lower than 99 slow down the playback speed compared to the speed of the original video.

1.4.3 NexEditor™ SDK ID Formats

The naming format of IDs identifies the type of the effects.

Theme ID

Format: companyname.kinemaster.themename

ex) com.nexstreaming.kinemaster.basic

Theme Effect ID

Theme effects included in a theme does not have any set effect ID. Because these clip effects are dependent on the themes, if the theme of a project changes, the bundled clip effects will also change accordingly automatically.

Format: companyname.kinemaster.themename.themteeffectype

ex) com.nexstreaming.kinemaster.basic.accent

Theme Effect Types:

- Accent : Clip effect applied on clips without text.
- Ending : Clip effect applied on the last clip of a project with text.
- Middle : Clip effect applied on clips with text that are not the first or last clip.
- Opening : Clip effect applied on the first clip with text.
- Transition : Transition effect applied between clips except for the first or last clip.

Basic effect ID

IDs of effects that are not dependent on any NexEditor™ SDK themes. Even if the theme changes, the independent clip effects will remain on the clips.

Format: companyname.kinemaster.builtinX.effecttype.effectname

ex) com.nexstreaming.kinemaster.builtin6.transition.flip

Effect Types:

- Accent : Clip effect without text.
- Title : Clip effect with text.
- Transition : Transition effect
- Overlay : Image effect

1.5 Application Porting Guide

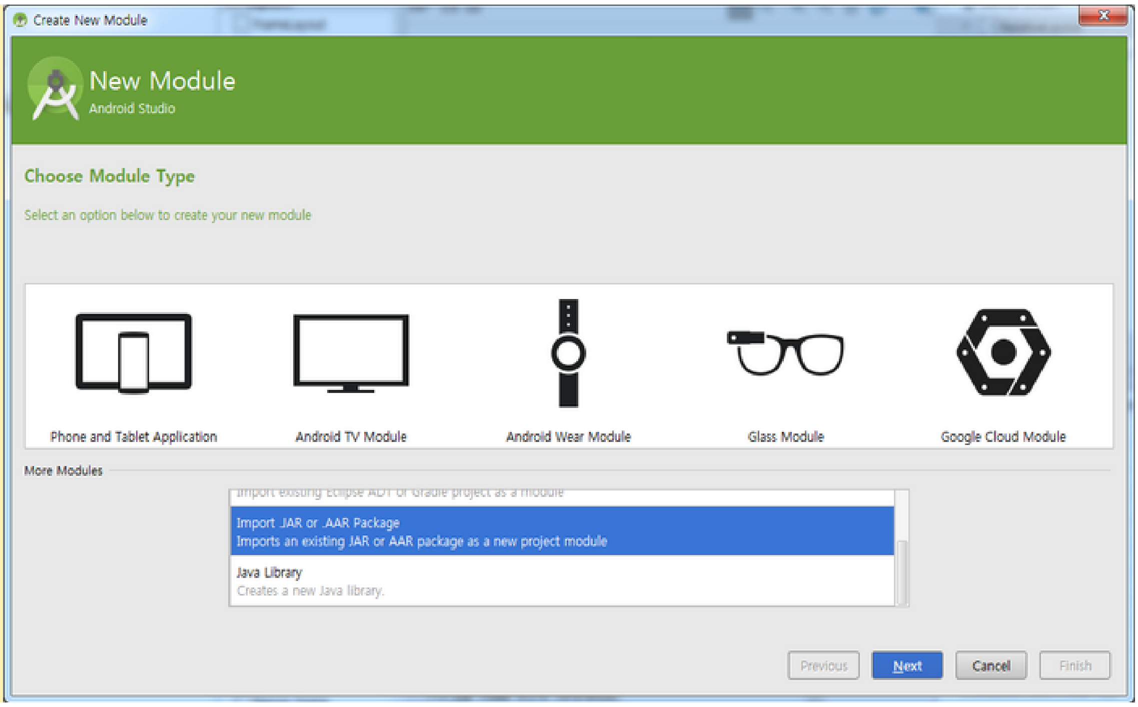
1.5.1 How to import the NexEditor™ SDK

For developer convenience, the SDK library and sample code for Android Studio and Eclipse are provided as packages in the SDK folder.

Android Studio, the official Android IDE(http://developer.android.com/sdk/index.html), is recommended for development with the NexEditor™ SDK.

For Android Studio

1. **Create a New Project for your Application.**
   (a) Warning : *The minSdkVersion (minimum API level) of the NexEditor™ SDK is 19 (KitKat). For compatibility, minSdkVersion has to be set to 19 when creating the project.*
2. **Add NexEditor™ SDK AAR Module:**
   (a) Select **File** > **New Module** from the Menu.
   (b) From the **New Module** window, select **Import .JAR** or **AAR Package**, then click **Next**.



Create New Module

New Module

Android Studio

Choose Module Type

Select an option below to create your new module

More Modules

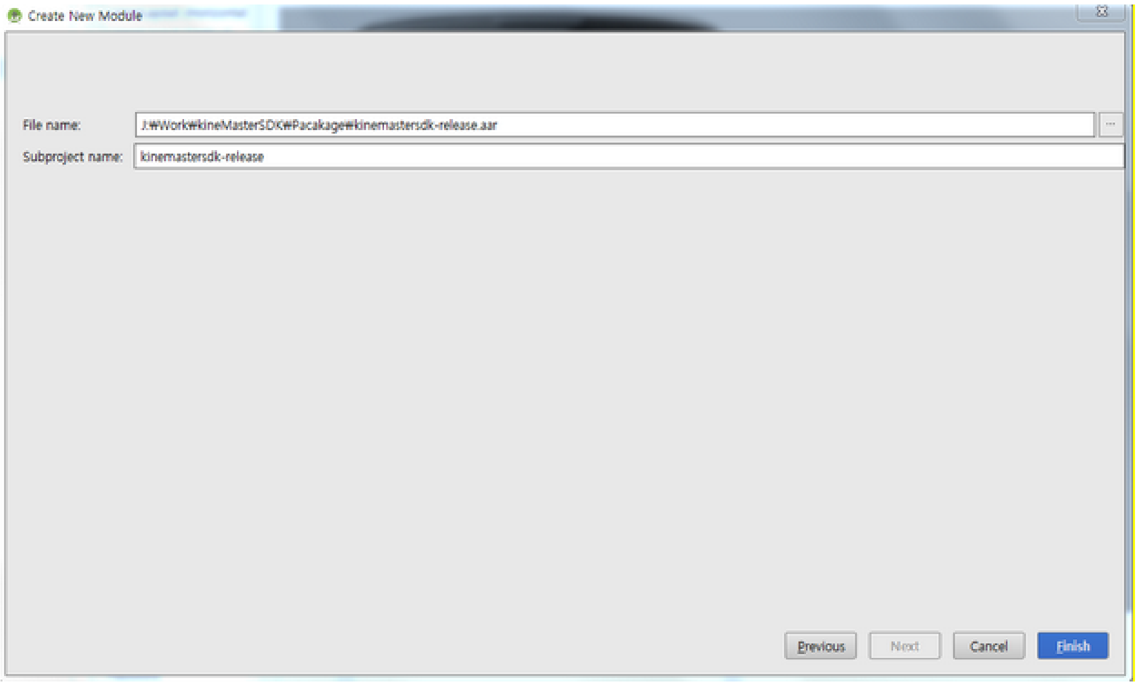
Import existing Eclipse ADT or Gradle project as a module

Import .JAR or .AAR Package
Imports an existing JAR or AAR package as a new project module

Java Library
Creates a new Java library.

Figure 1.4: Creating New Module

(c) Search for the file name '`nexeditorsdk-release.aar`'.



File name: J:₩Work₩kineMasterSDK₩Pacakage₩kinemastersdk-release.aar

Subproject name: kinemastersdk-release

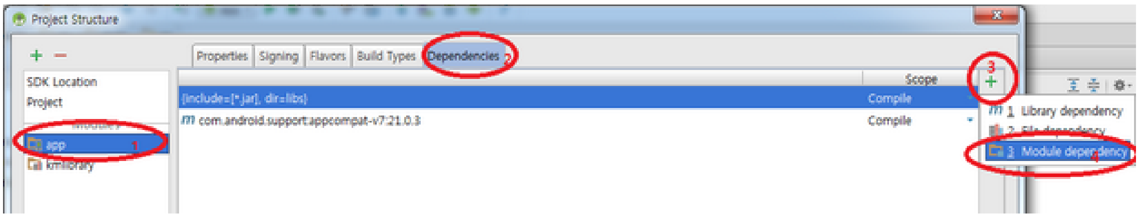
Figure 1.5: Creating New Module

(d) Finally, click the **Finish** button to successfully add the NexEditor™ SDK module.

3. **Set dependency to use the NexEditor™ SDK on the project.**

   (a) Select **File** > **Project Structure** from the Menu.

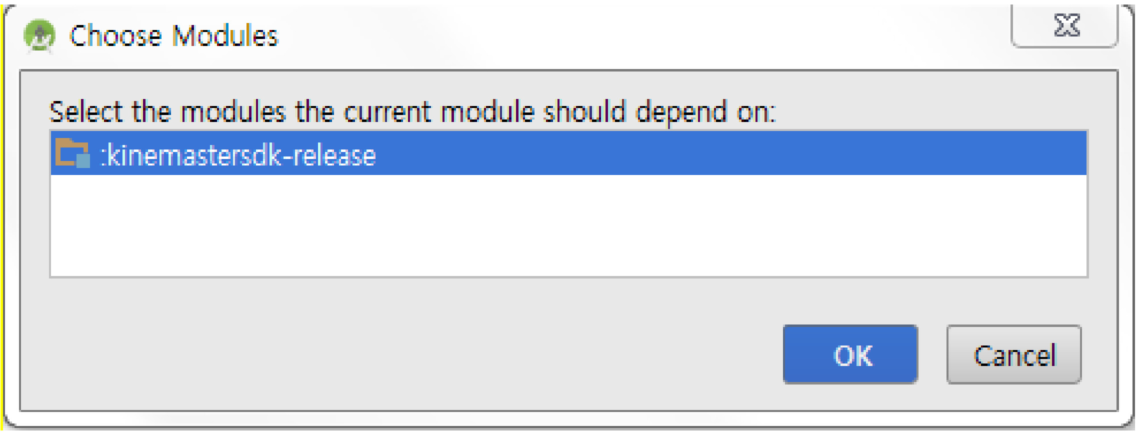
   (b) Follow the steps shown below to choose the module the application should depend on.



	Scope
{include=[*.jar], dir=libs}	Compile
com.android.support:appcompat-v7:21.0.3	Compile

Figure 1.6: Project Structure

(c) Select the NexEditor™ SDK module '`nexeditorsdk-release`' and add it to the dependency list.

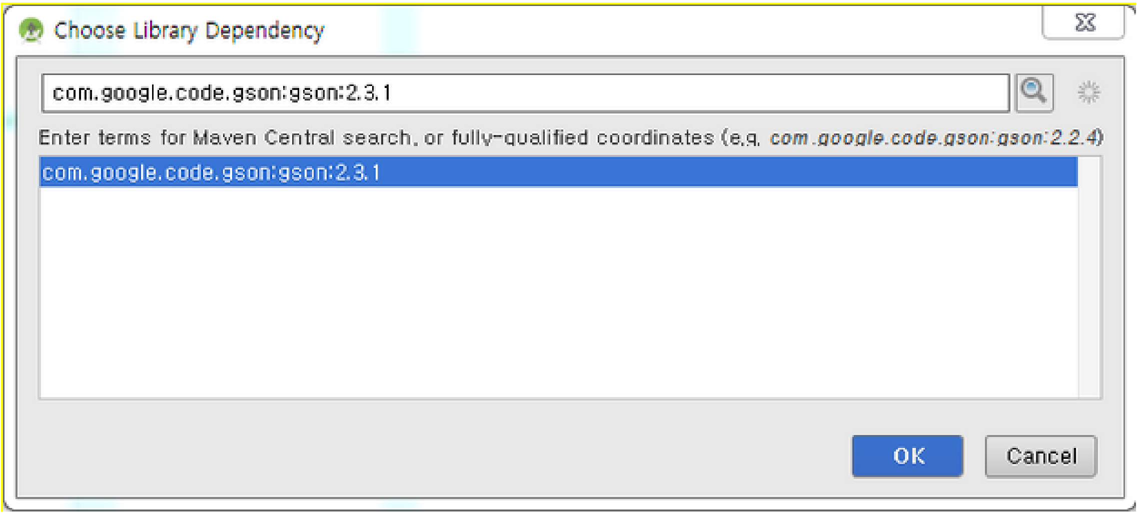


Select the modules the current module should depend on:

:kinemastersdk-release

Figure 1.7: Choose Modules

(d) The NexEditor™ SDK also requires *com.google.code.gson:gson:2.3.1.* Continued with the last step of 3.b, choose a **library dependency** to find *com.google.code.gson:gson:2.3.1* to add.



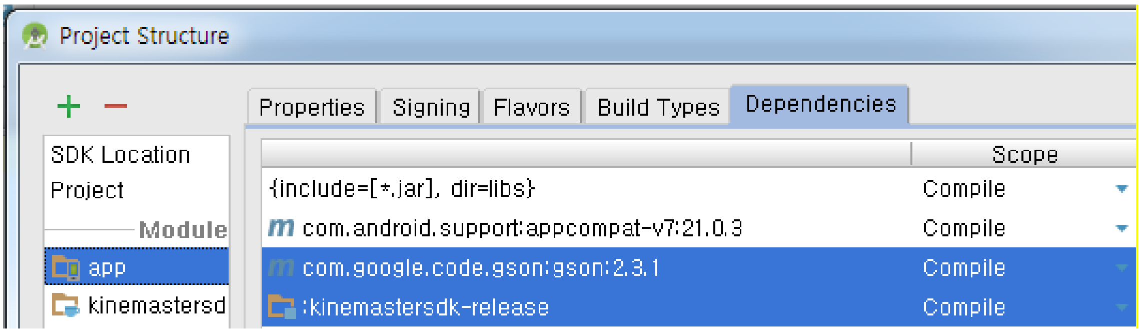
com.google.code.gson:gson:2.3.1

Enter terms for Maven Central search, or fully-qualified coordinates (e.g. *com.google.code.gson:gson:2.2.4*)

com.google.code.gson:gson:2.3.1

Figure 1.8: Choose Library Dependency

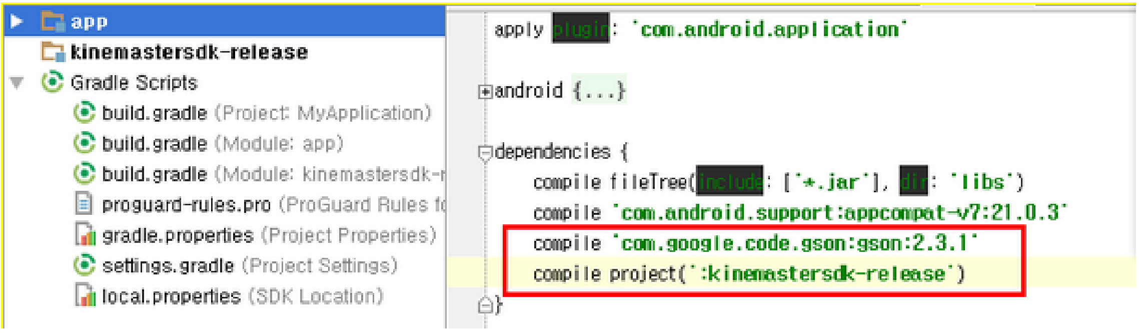
(e) Finally, if both *com.google.code.gson:gson:2.3.1* and /c 'nexeditorsdk-release' appear in the Dependencies list, the dependencies have been successfully set.



	Scope
{include=[*.jar], dir=libs}	Compile
m com.android.support:appcompat-v7:21.0.3	Compile
m com.google.code.gson:gson:2.3.1	Compile
:kinemastersdk-release	Compile

Figure 1.9: Project Structure

(f) Also, if the procedure above was successful, the code that appears below will be included in the `build.gradle` of the application.


app
kinemastersdk-release
Gradle Scripts
build.gradle (Project: MyApplication)
build.gradle (Module: app)
build.gradle (Module: kinemastersdk-r
proguard-rules.pro (ProGuard Rules f
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)

```
apply plugin: 'com.android.application'

android {...}

dependencies {
    compile fileTree(include: ['*.jar'], dir: 'libs')
    compile 'com.android.support:appcompat-v7:21.0.3'
    compile 'com.google.code.gson:gson:2.3.1'
    compile project(':kinemastersdk-release')
}
```

Figure 1.10: build.gradle code

For Eclipse

The instructions that follow are an example of a project `'EditorSDKDemo'` being created in the application.

Note that the `minSdkVersion` (minimum API level) of the NexEditor™ SDK is 19 (KitKat). For compatibility, `minSdkVersion` has to be set to 19 when creating the project.

1. Create a new project for your application.

   (a) Select **File** > **Project.**

   (b) Select **Android Application Project** then click **Next.**

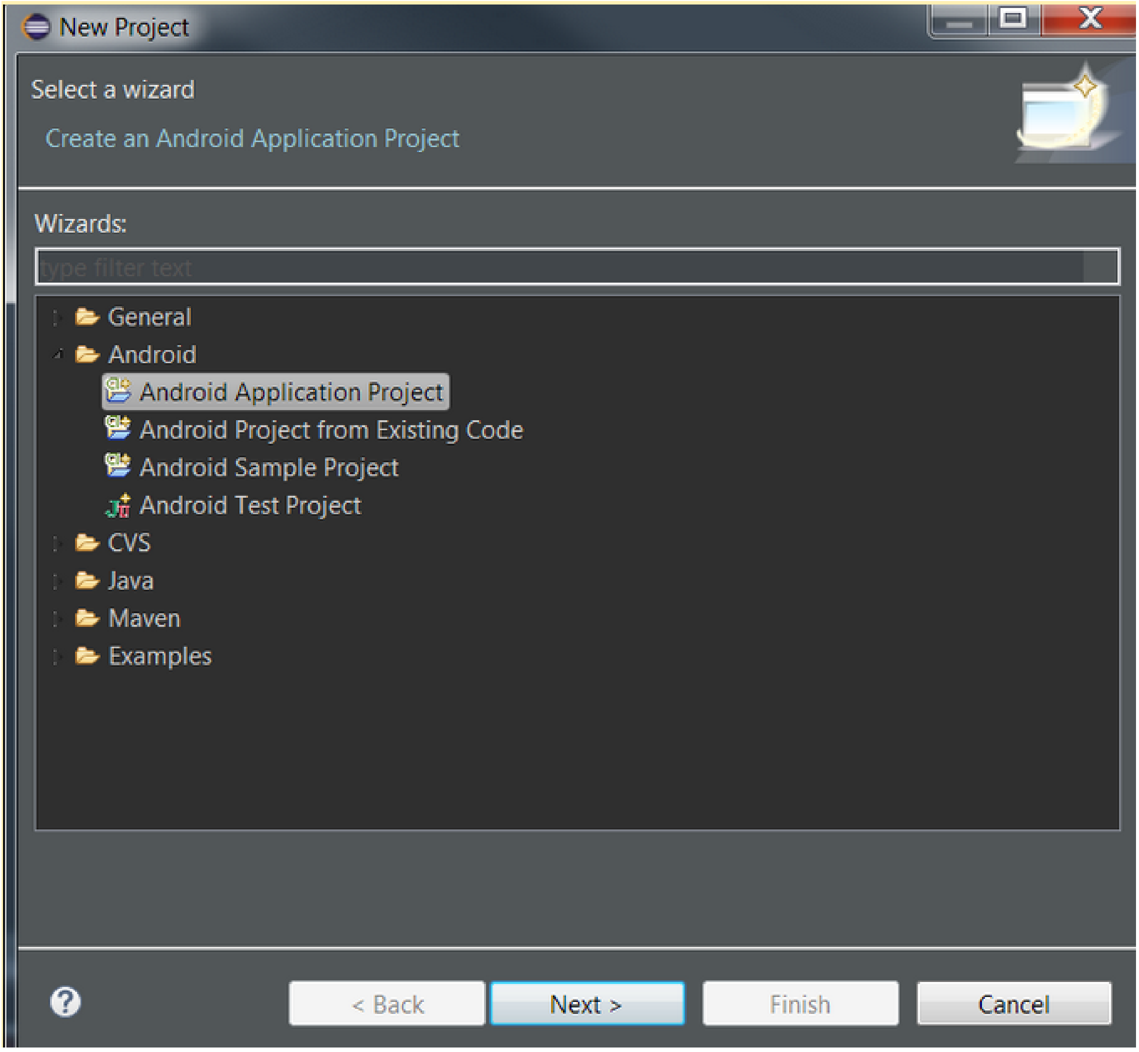
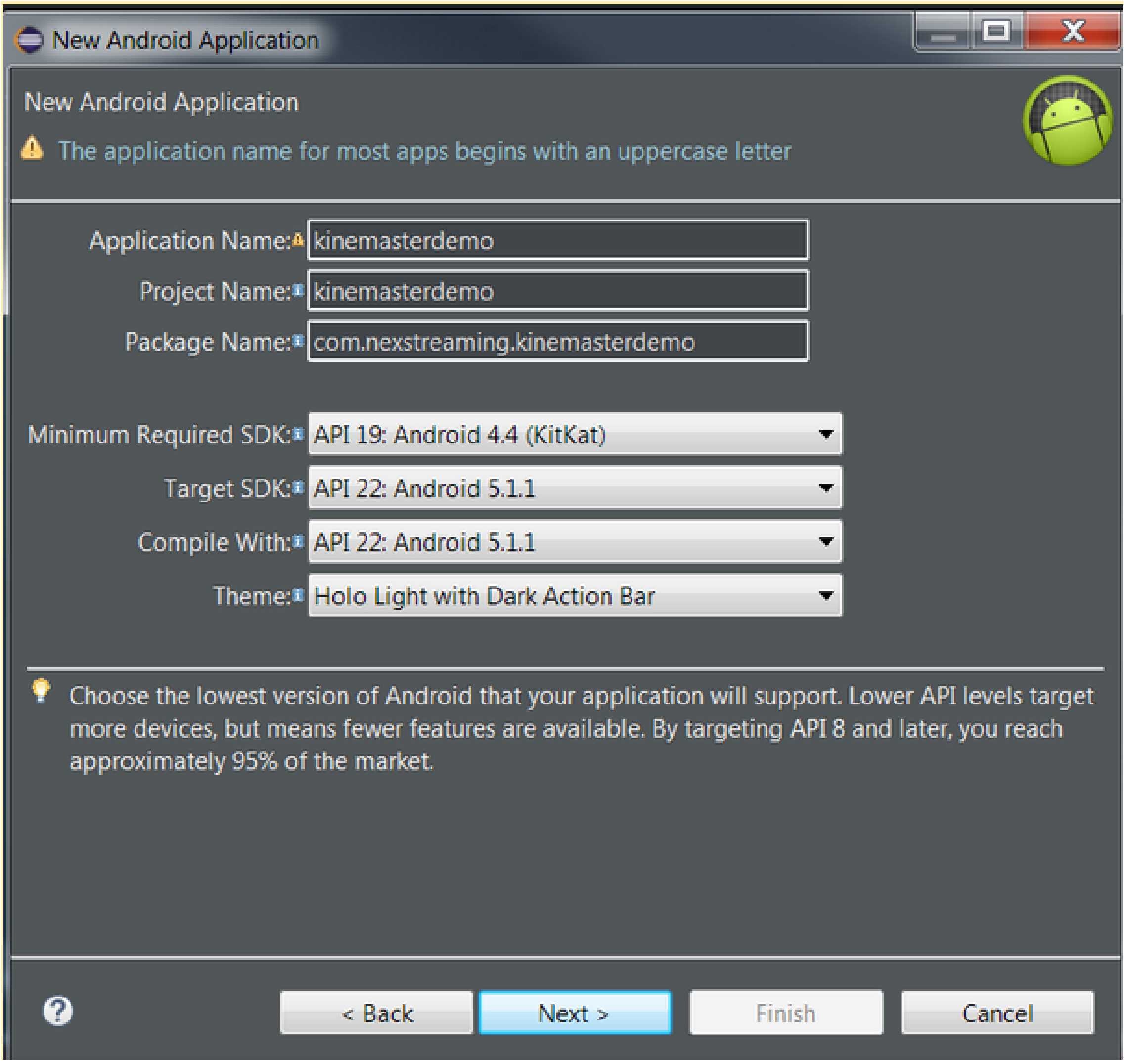


Figure 1.11: Creating New Project

   (c) For example, to make a project called `nexeditorDemo`, because the NexEditor™ SDK supports minSdkVersion 19(KitKat) or higher, **Minimum Required SDK:API 19:Android 4.4 (KitKat)** has to be selected.



New Android Application

⚠ The application name for most apps begins with an uppercase letter

Application Name: kinemasterdemo

Project Name: kinemasterdemo

Package Name: com.nexstreaming.kinemasterdemo

Minimum Required SDK: API 19: Android 4.4 (KitKat)

Target SDK: API 22: Android 5.1.1

Compile With: API 22: Android 5.1.1

Theme: Holo Light with Dark Action Bar

Choose the lowest version of Android that your application will support. Lower API levels target more devices, but means fewer features are available. By targeting API 8 and later, you reach approximately 95% of the market.

Figure 1.12: New Android Application

2. In order to use the NexEditor™ SDK, register the provided project as a Library.
   (a) Select **File** > **Import.**
   (b) Select **Existing Android Code Into Workspace** and then click **Next.**

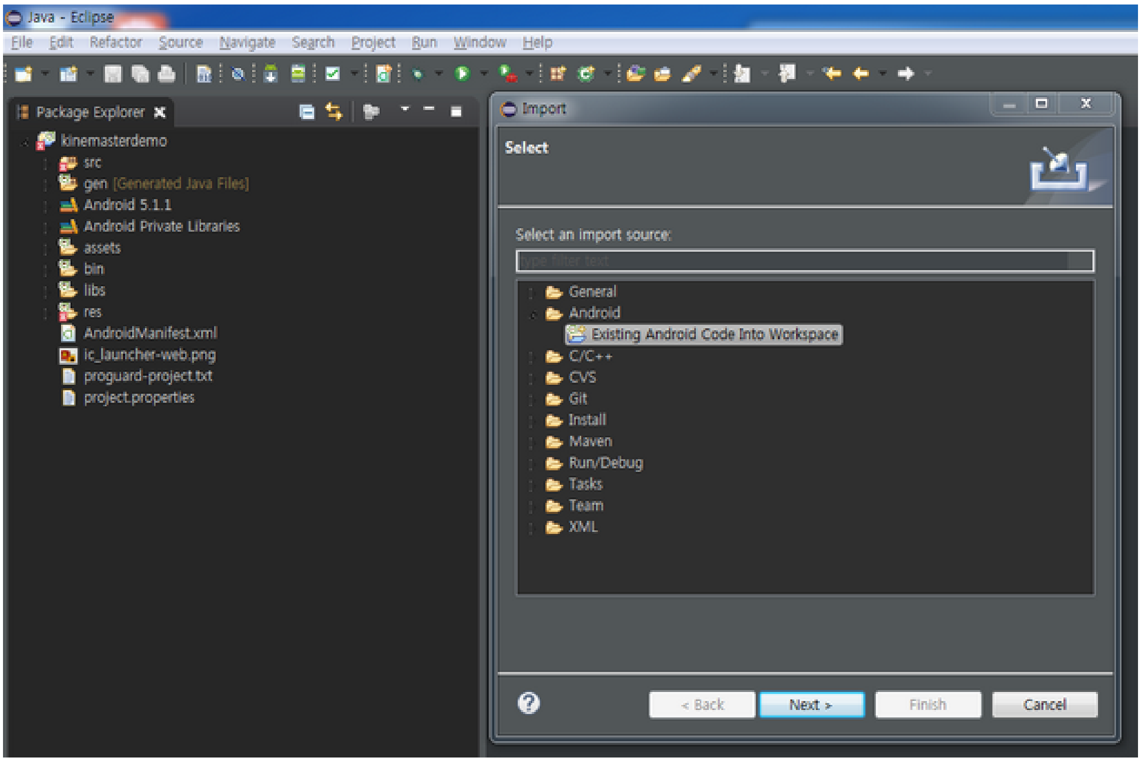
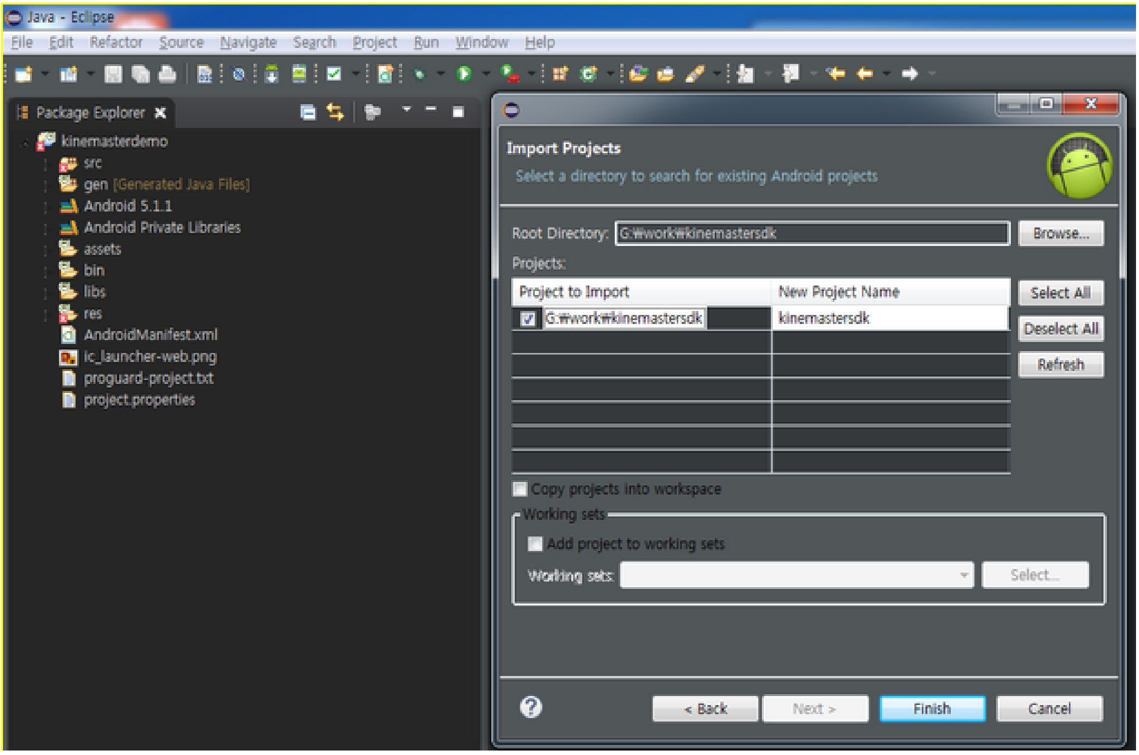


Figure 1.13: Selecting Existing Android Code

(c) Select a project path called `nexeditorsdk` and then click **Finish**.



Import Projects

Select a directory to search for existing Android projects

Root Directory: G:₩work₩kinemastersdk

Projects:

Project to Import	New Project Name
☑ G:₩work₩kinemastersdk	kinemastersdk

□ Copy projects into workspace

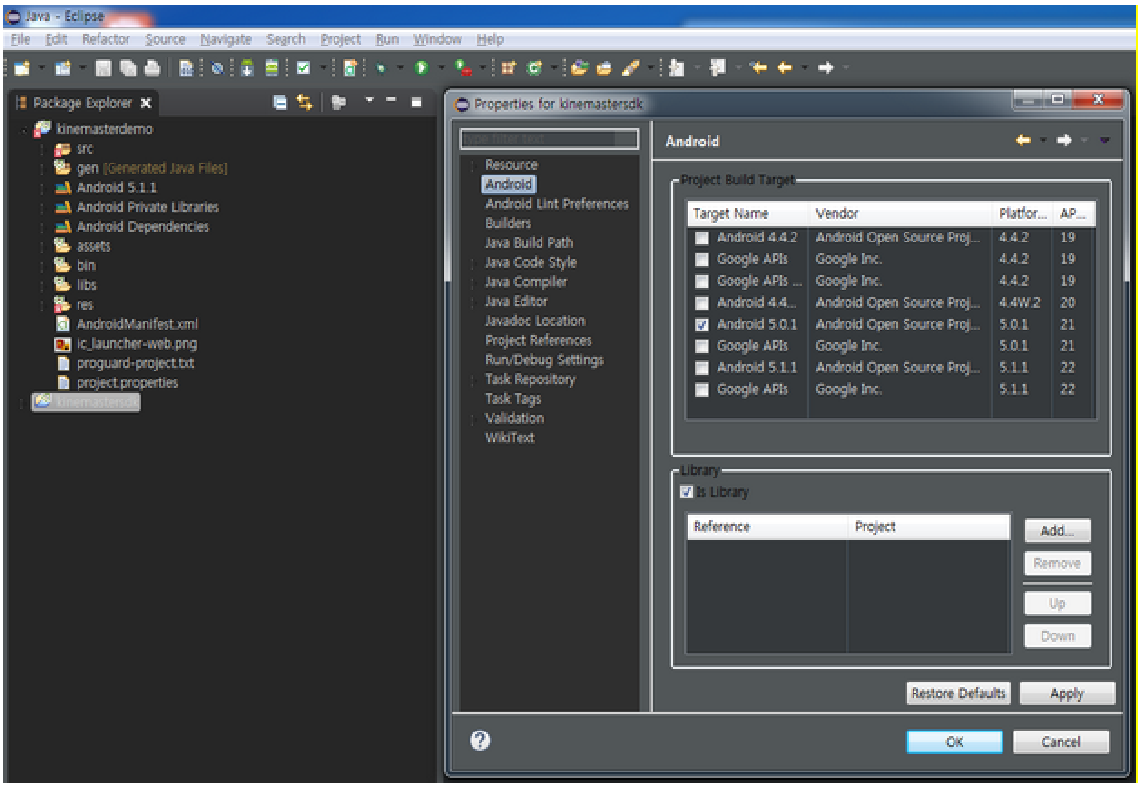
Working sets

□ Add project to working sets

Working sets:

Figure 1.14: Selecting `nexeditorsdk`

(d) Check on the **Package Explorer** tab to see if the project `nexeditorsdk` was created. Right click on `nexeditorsdk` and select **Properties** > **Android** to see whether ‘is Library’ from the Library section is checked.



Android

Project Build Target

Target Name	Vendor	Platfor...	AP...
□ Android 4.4.2	Android Open Source Proj...	4.4.2	19
□ Google APIs	Google Inc.	4.4.2	19
□ Google APIs ...	Google Inc.	4.4.2	19
□ Android 4.4...	Android Open Source Proj...	4.4W.2	20
☑ Android 5.0.1	Android Open Source Proj...	5.0.1	21
□ Google APIs	Google Inc.	5.0.1	21
□ Android 5.1.1	Android Open Source Proj...	5.1.1	22
□ Google APIs	Google Inc.	5.1.1	22

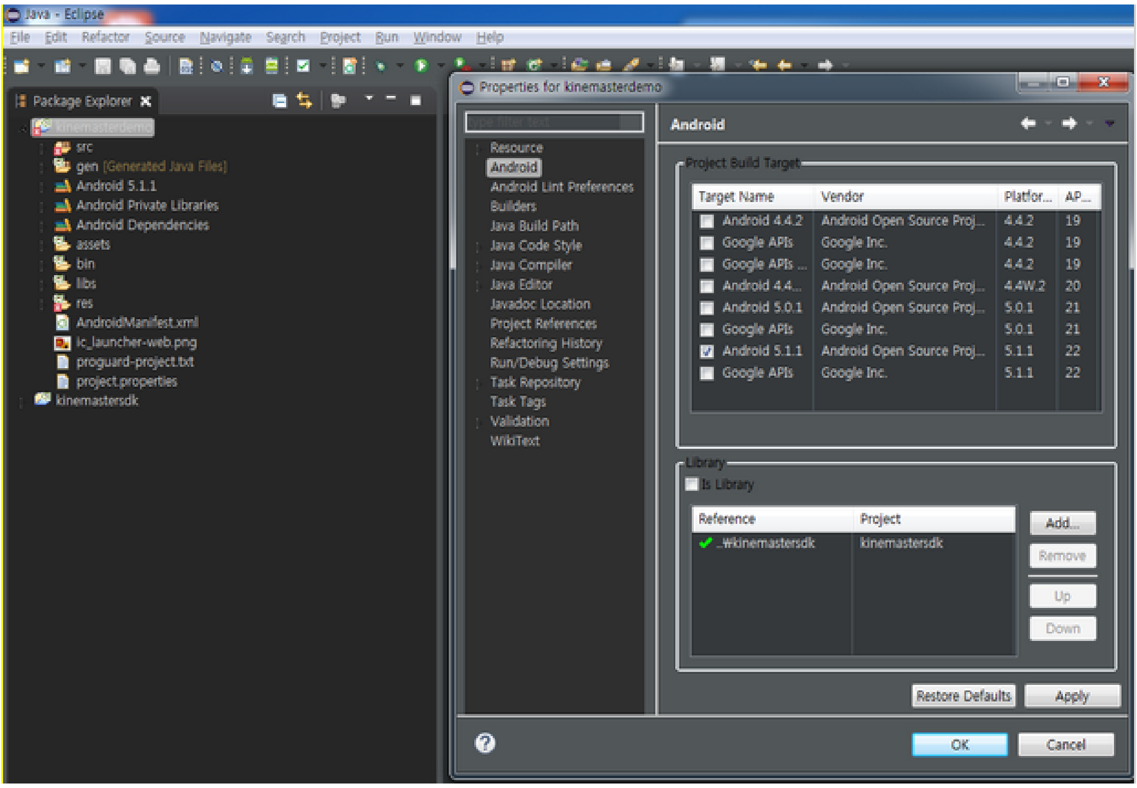
Library

☑ Is Library

Reference	Project

Figure 1.15: Checking Package Explorer Tab

(e) From the **Package Explorer tab**, right click on `nexeditordemo` project to select **Properties** > **Android.** Click on **Add...** to add the `nexeditorsdk` project to the Library list and then click **OK**.



Android

Project Build Target

Target Name	Vendor	Platfor...	AP...
□ Android 4.4.2	Android Open Source Proj...	4.4.2	19
□ Google APIs	Google Inc.	4.4.2	19
□ Google APIs ...	Google Inc.	4.4.2	19
□ Android 4.4...	Android Open Source Proj...	4.4W.2	20
□ Android 5.0.1	Android Open Source Proj...	5.0.1	21
□ Google APIs	Google Inc.	5.0.1	21
☑ Android 5.1.1	Android Open Source Proj...	5.1.1	22
□ Google APIs	Google Inc.	5.1.1	22

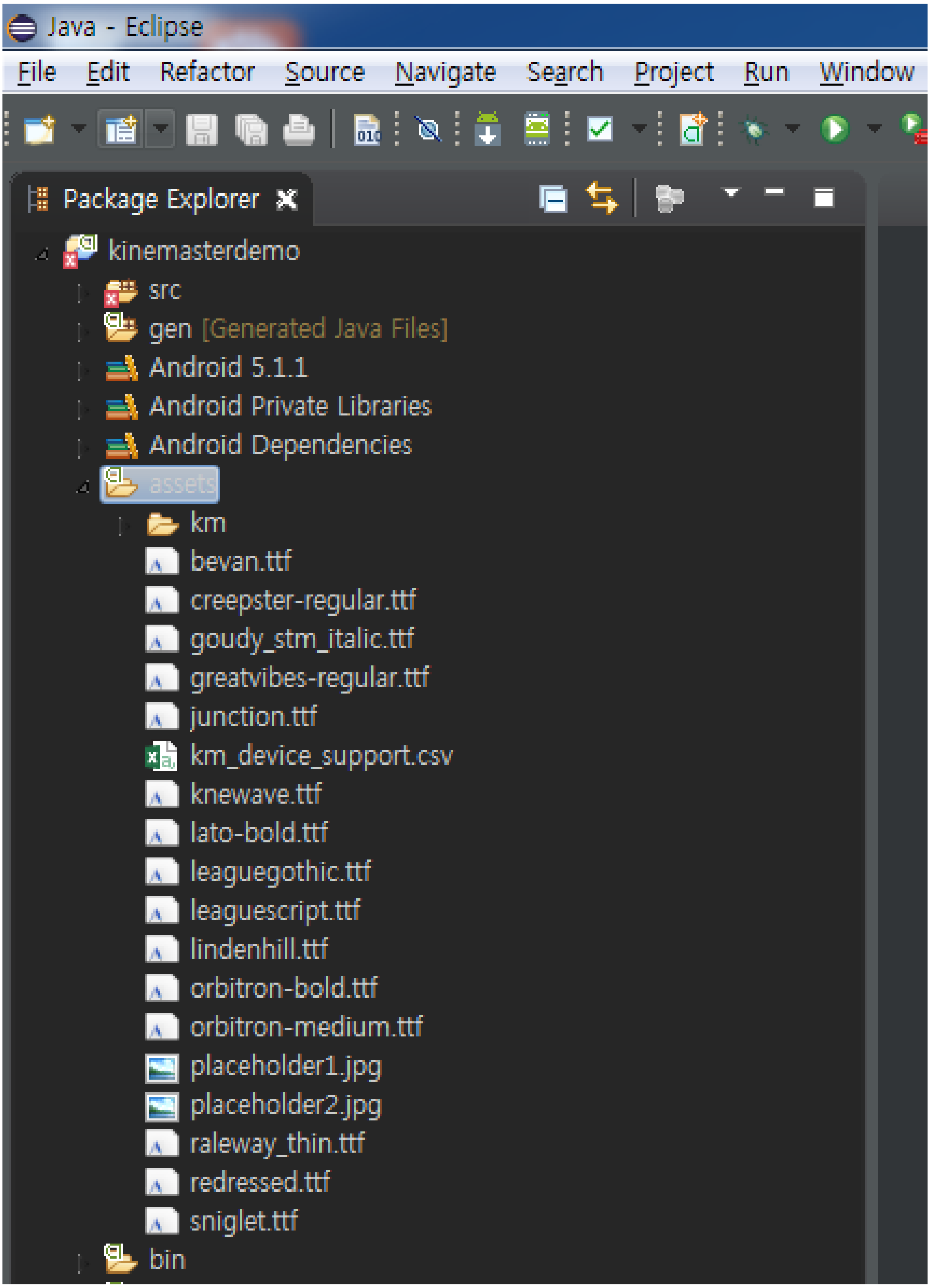
Library

□ Is Library

Reference	Project
✓ ..₩kinemastersdk	kinemastersdk

Figure 1.16: Adding the `nexeditorsdk` Project to the Library List

(f) To use themes and transitions, copy provided asset contents into the `assets` folder in the `nexeditordemo` project.


Java - Eclipse
File Edit Refactor Source Navigate Search Project Run Window
Package Explorer

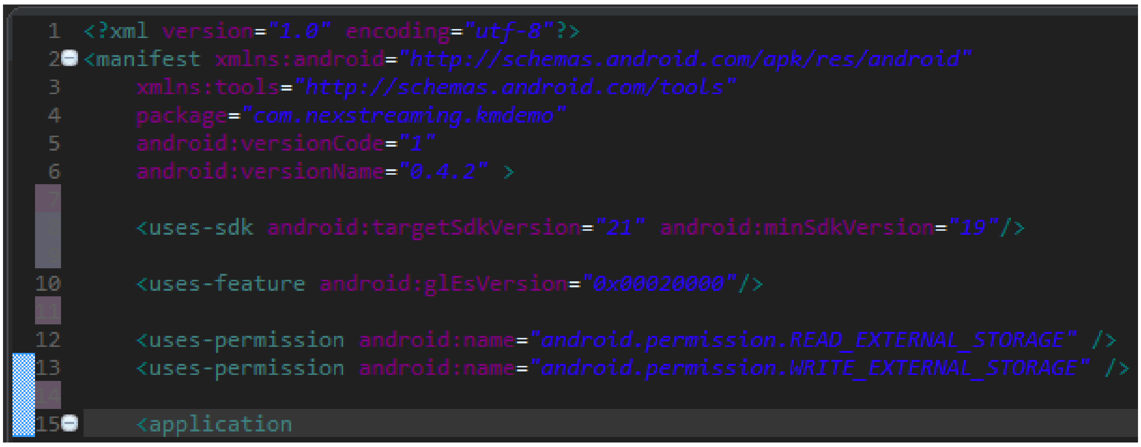
```
kinemasterdemo
    src
    gen [Generated Java Files]
    Android 5.1.1
    Android Private Libraries
    Android Dependencies
    assets
        km
        bevan.ttf
        creepster-regular.ttf
        goudy_stm_italic.ttf
        greatvibes-regular.ttf
        junction.ttf
        km_device_support.csv
        knewave.ttf
        lato-bold.ttf
        leaguegothic.ttf
        leaguescript.ttf
        lindenhill.ttf
        orbitron-bold.ttf
        orbitron-medium.ttf
        placeholder1.jpg
        placeholder2.jpg
        raleway_thin.ttf
        redressed.ttf
        sniglet.ttf
    bin
```

Figure 1.17: Copying Asset Contents into the `assets` Folder

(g) OpenGL ES 2.0 and the permissions listed below should be stated in the 2.7 `AndroidManifest.xml` file.

```
<uses-feature android:glEsVersion="0x00020000" />

<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```



```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.nexstreaming.kmdemo"
    android:versionCode="1"
    android:versionName="0.4.2" >

    <uses-sdk android:targetSdkVersion="21" android:minSdkVersion="19"/>

    <uses-feature android:glEsVersion="0x00020000"/>

    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

    <application
```

Figure 1.18: `AndroidManifest`.xml File

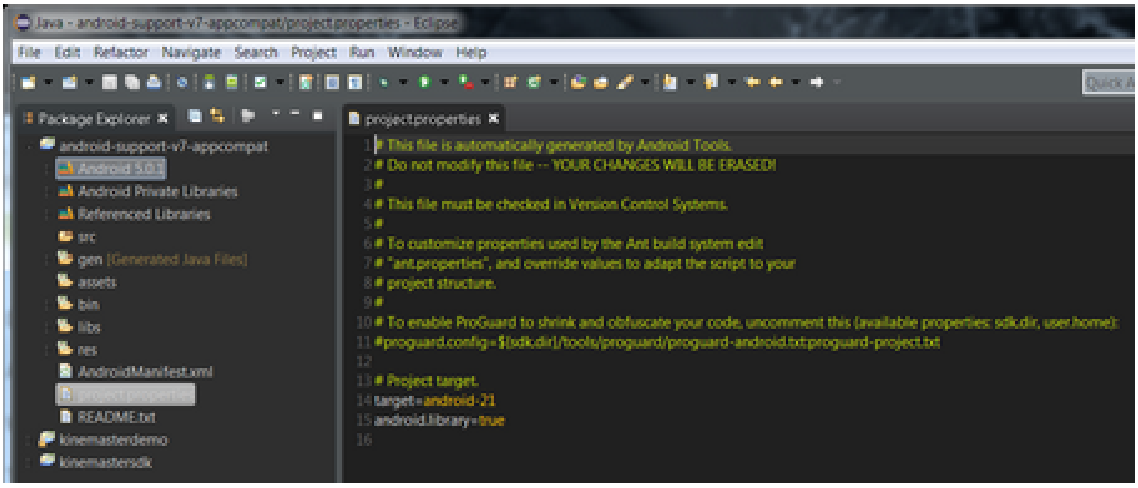
1.5.2 Porting Android Studio Application Source to Eclipse

If the project `EditorSDKDEMO` (located in `/sample/as`) is run with Eclipse, the APK will not build properly because `EditorSDKDEMO` is constructed based on the Android Studio Tool. To run the `EditorSDKDEMO` project in Eclipse, please follow the steps outlined below.

NOTE: These instructions are written based on the supposition that the example project created is called `nexeditordemo`.

The DemoApp project uses the v 7 Appcompat Library provided by Android. Therefore, the project (for the library) should be created, referring to the flow information provided at: <u>`http://developer.android.com/tools/support-library/setup.html`<u> > **Adding Libraries with Resources** > **Using Eclipse**, to use the current project as a library in the `nexeditordemo` project.

1. Select **File** > **Import.**
2. Select **Existing Android Code** from the workspace list.
3. Search for the `/extras/android/support/v7/appcompat` folder in the Android SDK directory.
4. Click **Finish** to import the project.
5. Open up the file `project.properties` to check for `#Project target`. If `target` is set to `=android-17`, change it to `=android-21`.



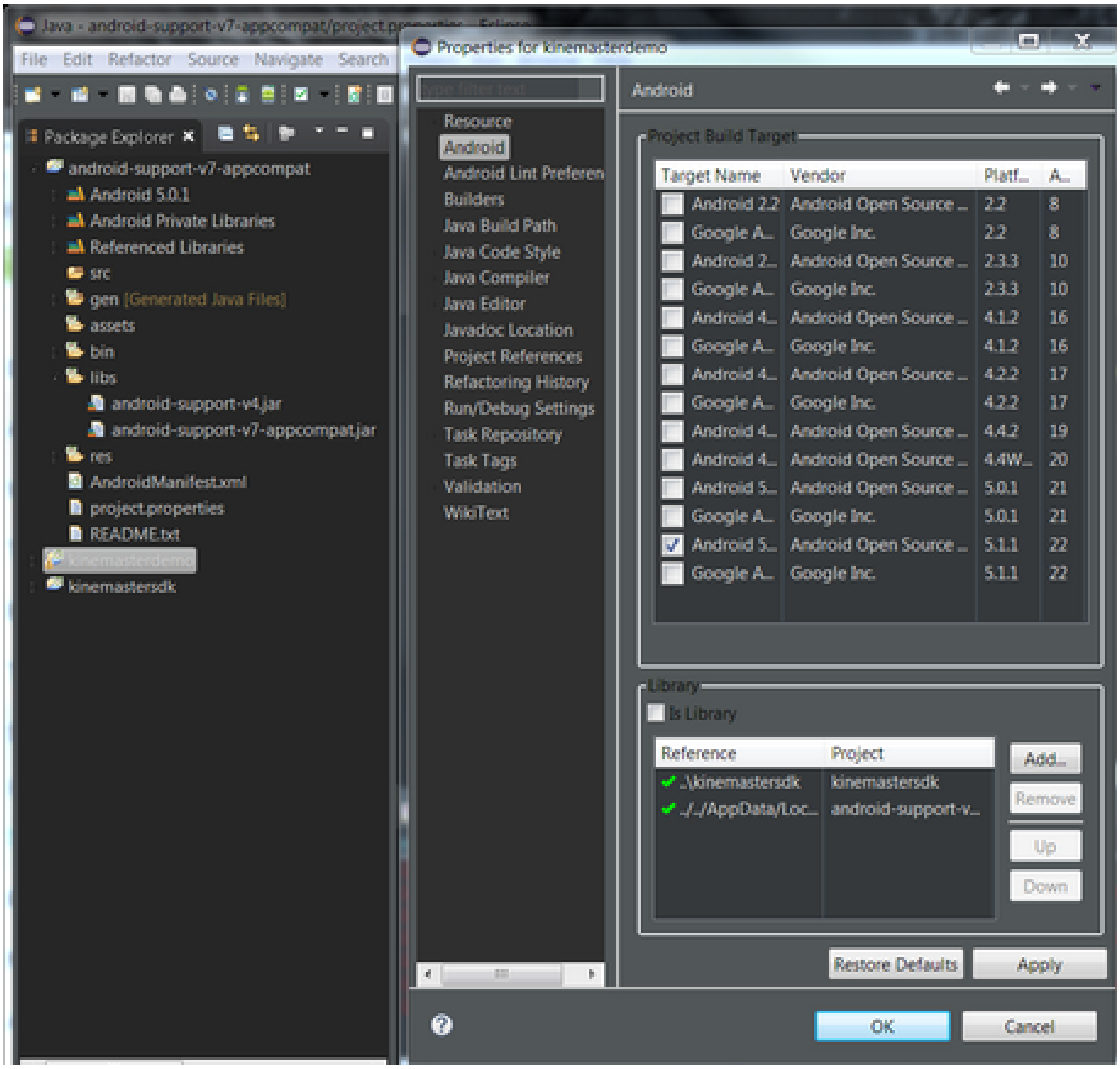
Java - android-support-v7-appcompat/project.properties - Eclipse
File Edit Refactor Navigate Search Project Run Window Help
Quick A
Package Explorer
android-support-v7-appcompat
Android 5.0.1
Android Private Libraries
Referenced Libraries
src
gen [Generated Java Files]
assets
bin
libs
res
AndroidManifest.xml
project.properties
README.txt
kinemasterdemo
kinemastersdk
project.properties

```
# This file is automatically generated by Android Tools.
# Do not modify this file -- YOUR CHANGES WILL BE ERASED!
#
# This file must be checked in Version Control Systems.
#
# To customize properties used by the Ant build system edit
# "ant.properties", and override values to adapt the script to your
# project structure.
#
# To enable ProGuard to shrink and obfuscate your code, uncomment this (available properties: sdk.dir, user.home):
#proguard.config=${sdk.dir}/tools/proguard/proguard-android.txt:proguard-project.txt

# Project target.
target=android-21
android.library=true
```

Figure 1.19: Checking `project.properties` File

6. Right click on the `nexeditordemo` project to select **Properties** > **Android** and open a window. From the window, click on **Add...** to add project `android-support-v7-appcompat` to the Library list, then click **OK**.



	Target Name	Vendor	Platf...	A...
□	Android 2.2	Android Open Source ...	2.2	8
□	Google A...	Google Inc.	2.2	8
□	Android 2...	Android Open Source ...	2.3.3	10
□	Google A...	Google Inc.	2.3.3	10
□	Android 4...	Android Open Source ...	4.1.2	16
□	Google A...	Google Inc.	4.1.2	16
□	Android 4...	Android Open Source ...	4.2.2	17
□	Google A...	Google Inc.	4.2.2	17
□	Android 4...	Android Open Source ...	4.4.2	19
□	Android 4...	Android Open Source ...	4.4W...	20
□	Android 5...	Android Open Source ...	5.0.1	21
□	Google A...	Google Inc.	5.0.1	21
☑	Android 5...	Android Open Source ...	5.1.1	22
□	Google A...	Google Inc.	5.1.1	22

□ Is Library

Reference	Project
✓ ..\kinemastersdk	kinemastersdk
✓ ../../AppData/Loc...	android-support-v...

Figure 1.20: Adding project `android-support-v7-appcompat` to the Library List

7. Delete all folders in the `nexeditordemo/res` folder.

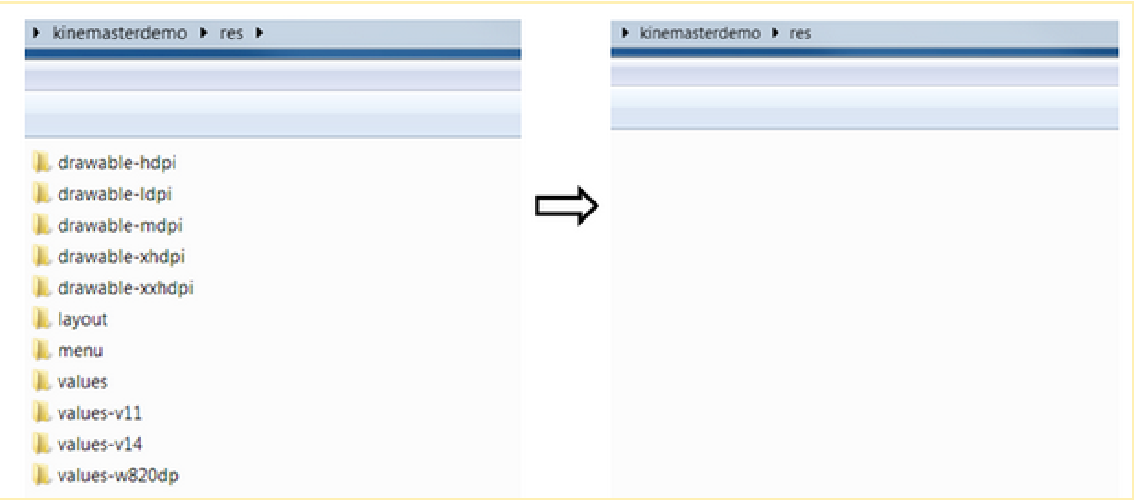


Figure 1.21: Deleting all Folders

8. Copy all the folders from `EditorSDKDEMO/sample/android_studio/src/main/res` into the `nexeditordemo/res` folder.

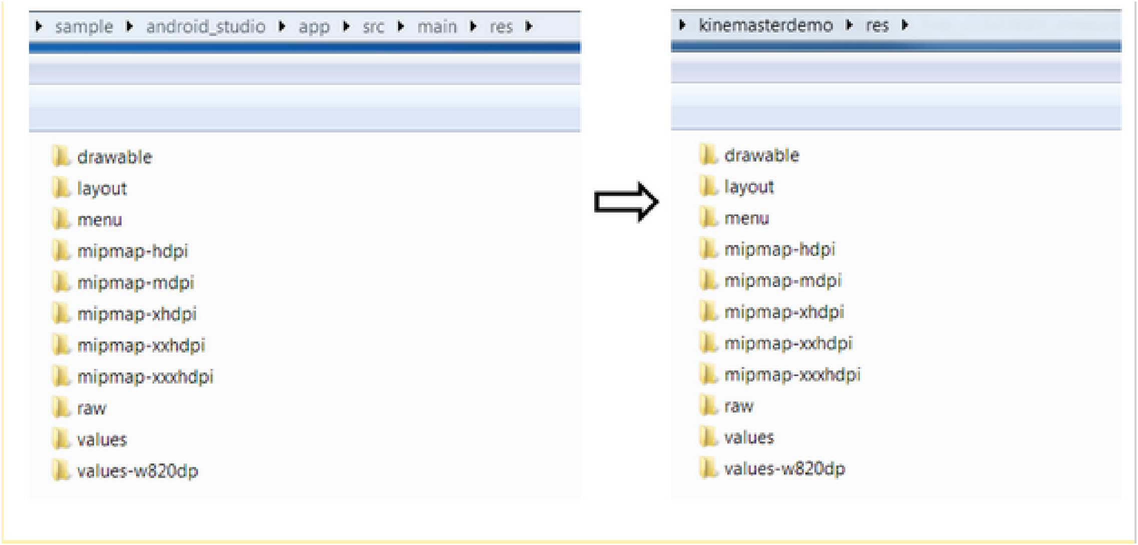


Figure 1.22: Copying all Folders

9. Delete all the folders in the `nexeditordemo/src` folder and then copy all the folders from `EditorSDKDEMO/sample/android_studio/main/java` into the `nexeditordemo/src` folder.

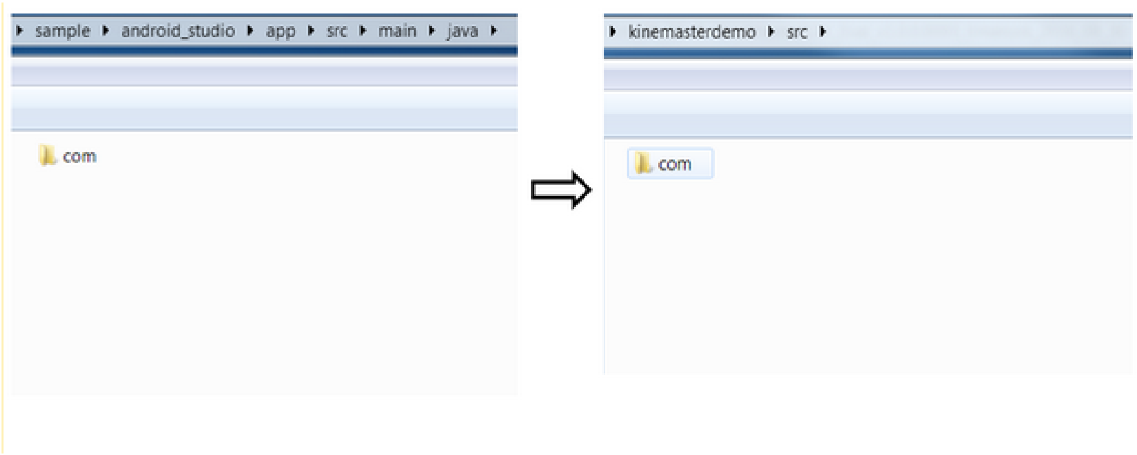


Figure 1.23: Deleting all Folders

10. Copy the two .jar files from `EditorSDKDEMO/sample/android_studio/app/libs` into the `nexeditordemo/libs` folder.

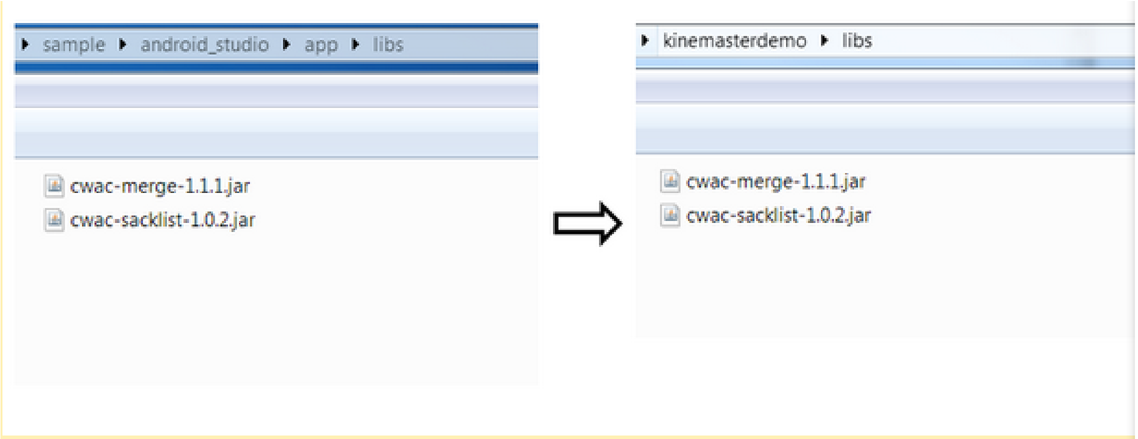


Figure 1.24: Copying two .jar Files

11. Open the file `nexeditordemo/`.settings/org.eclipse.jdt.core.prefs with the Editor Tool to edit the following values to **1.7** and save:
    - `org.eclipse.jdt.core.compiler.codegen.targetPlatform=1.7`
    - `org.eclipse.jdt.core.compiler.compliance=1.7`
    - `org.eclipse.jdt.core.compiler.source=1.7`

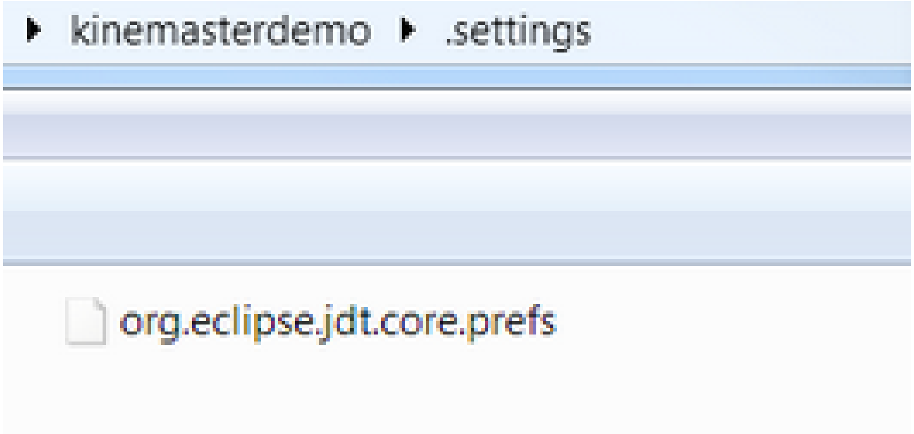


Figure 1.25: Opening `org.eclipse.jdt.core.prefs` File

12. Copy the **AndroidManifest.xml** file from `EditorSDKDEMO/sample/android_studio/app/src/main` into the `nexeditordemo` folder.

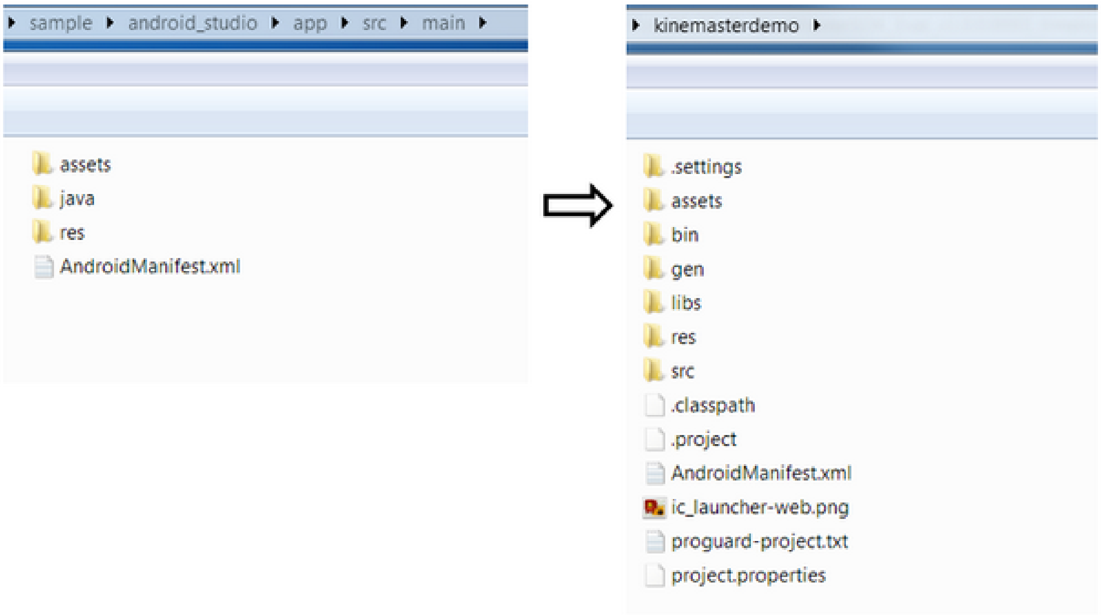


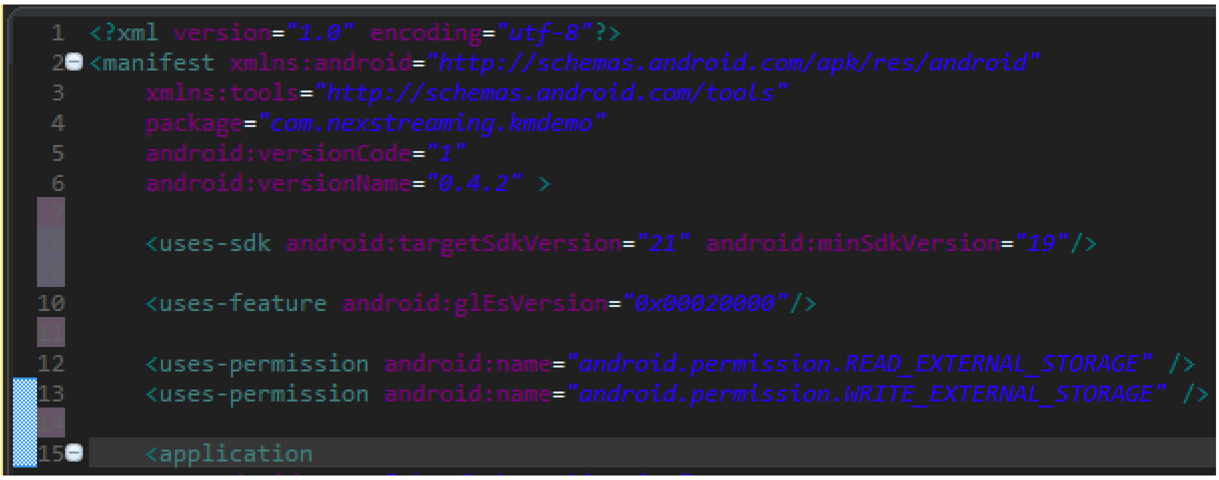
Figure 1.26: Copying `AndroidManifest.xml` File

13. Include the following OpenGL ES 2.0 and other permissions in the file `AndroidManifest.xml`.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
   package="com.nexstreaming.editorsdkdemo"
   android:versionCode="1"
   android:versionName="0.4.2" >

   <uses-sdk android:targetSdkVersion=”21” android:minSdkVersion="19" />

   <uses-feature android:glEsVersion="0x00020000" />
   <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
   <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.nexstreaming.kmdemo"
    android:versionCode="1"
    android:versionName="0.4.2" >

    <uses-sdk android:targetSdkVersion="21" android:minSdkVersion="19"/>

    <uses-feature android:glEsVersion="0x00020000"/>

    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

    <application
```

Figure 1.27: `AndroidManifest.xml` width

1. Finally, try building the `nexeditordemo` project again to create the APK successfully.

Precautions

When using the Eclipse Tool, select marked icon on *figure 1.28* to open **Android SDK Manager Tool**.

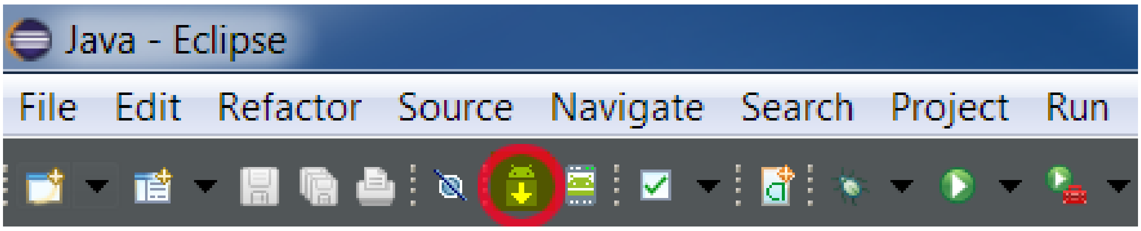
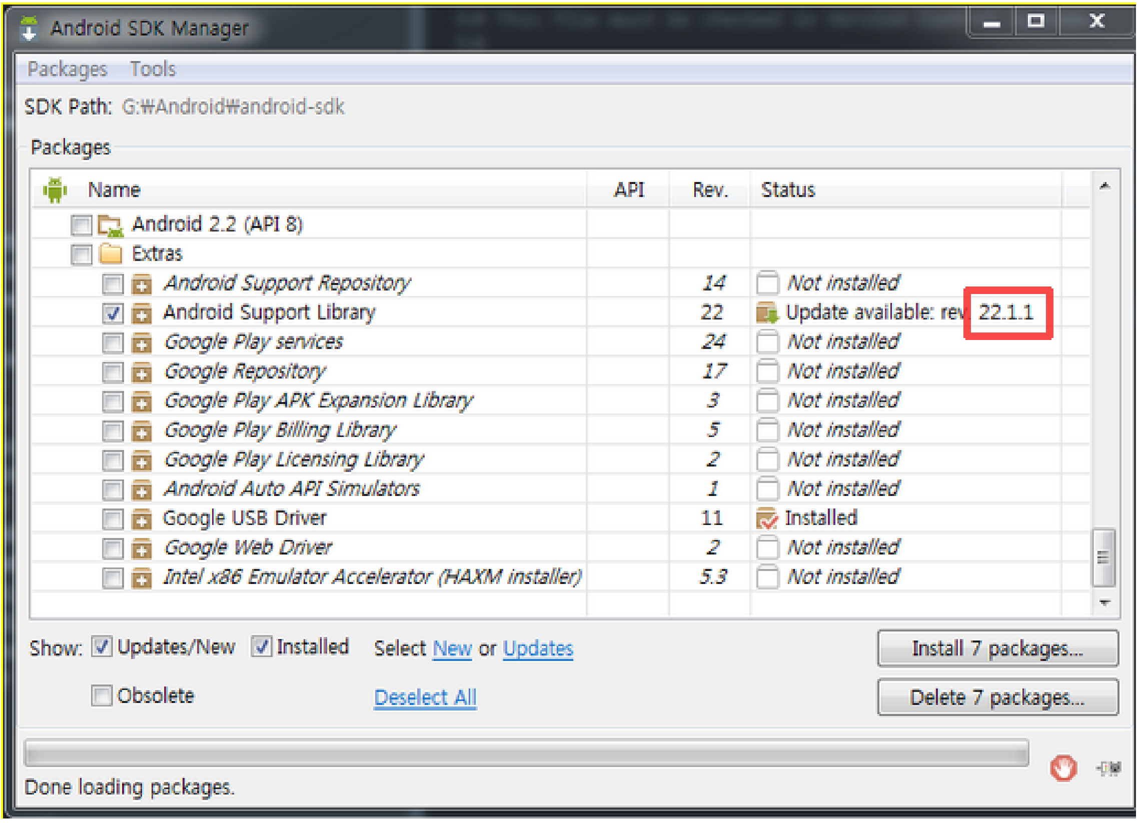


Figure 1.28: Android SDK Manager Tool Icon in Eclipse



SDK Path: G:₩Android₩android-sdk

Packages

Name	API	Rev.	Status
☐ Android 2.2 (API 8)			
☐ Extras			
☐ *Android Support Repository*		*14*	*Not installed*
☑ Android Support Library		22	Update available: rev. 22.1.1
☐ *Google Play services*		*24*	*Not installed*
☐ *Google Repository*		*17*	*Not installed*
☐ *Google Play APK Expansion Library*		*3*	*Not installed*
☐ *Google Play Billing Library*		*5*	*Not installed*
☐ *Google Play Licensing Library*		*2*	*Not installed*
☐ *Android Auto API Simulators*		*1*	*Not installed*
☐ Google USB Driver		11	Installed
☐ *Google Web Driver*		*2*	*Not installed*
☐ *Intel x86 Emulator Accelerator (HAXM installer)*		*5.3*	*Not installed*

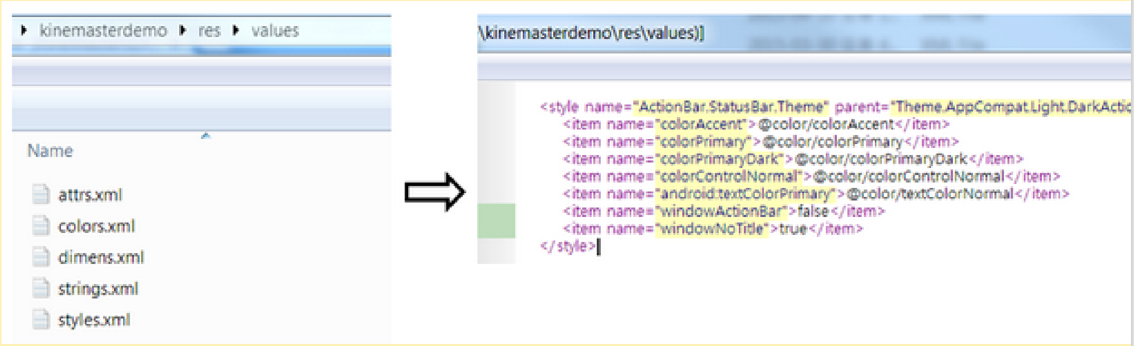
Show: ☑ Updates/New ☑ Installed Select New or Updates [Install 7 packages...]

☐ Obsolete Deselect All [Delete 7 packages...]

Done loading packages.

Figure 1.29: Android SDK Manager Tool

Check on the Android SDK Manager window, if the **Rev.** value of **Android Support Library** is 22.1.1, open the `styles.xml` file from the `nexeditordemo/res/value` folder to add *"<item name="window-NoTitle">true</item>"* into *<style name="ActionBar.StatusBar.Theme" parent="Theme.AppCompat.Light.Dark-ActionBar">*.



```
<style name="ActionBar.StatusBar.Theme" parent="Theme.AppCompat.Light.DarkActio
    <item name="colorAccent">@color/colorAccent</item>
    <item name="colorPrimary">@color/colorPrimary</item>
    <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
    <item name="colorControlNormal">@color/colorControlNormal</item>
    <item name="android:textColorPrimary">@color/textColorNormal</item>
    <item name="windowActionBar">false</item>
    <item name="windowNoTitle">true</item>
</style>
```

Figure 1.30: `styles.xml`

1.6 NexEditor™ SDK Structure and Implementation

1.6.1 Structure

NexEditor™ SDK Structure Diagram

Figure 1.31 is a diagram of the structure of the NexEditor™ SDK. The NexEditor™ SDK is primarily divided into `nexEffectLibrary` which manages the effects and `nexEngine` which manages the clips and actual functions.

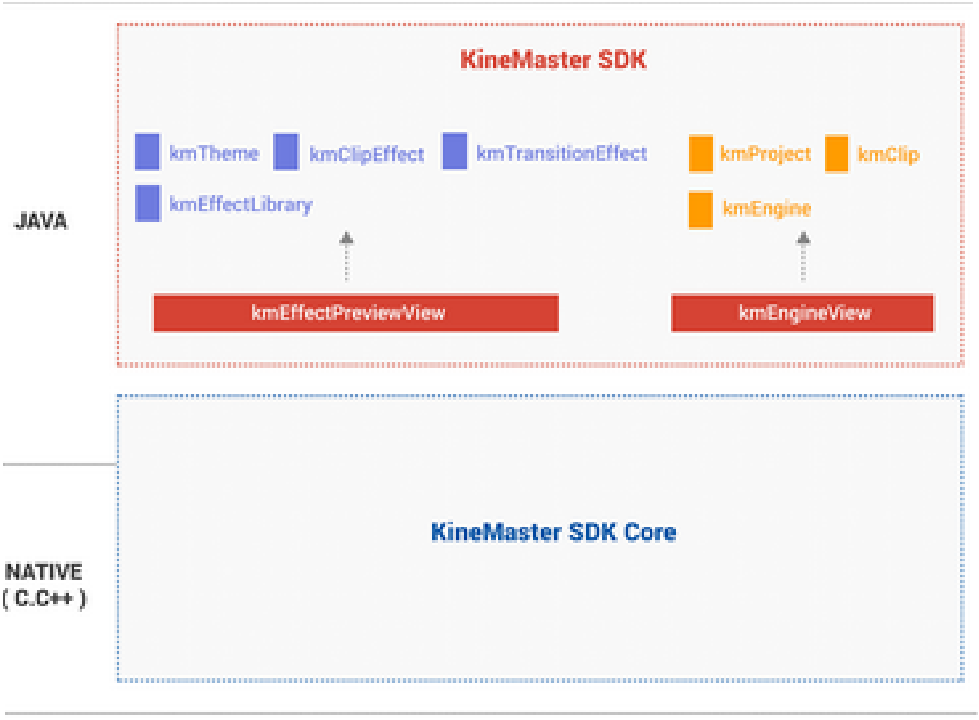


Figure 1.31: NexEditor SDK Structure Diagram

NexEditor™ SDK Flow Chart

Figure 1.32 is a flow chart of the NexEditor™ SDK. `nexClip` generates clips which will be put into `nexPorject` to be edited. Edited `nexProject` projects will be passed into `nexEngin` to be previewed or exported. Care must be taken to add `nexClip` clips into a `nexProject` to edit them.

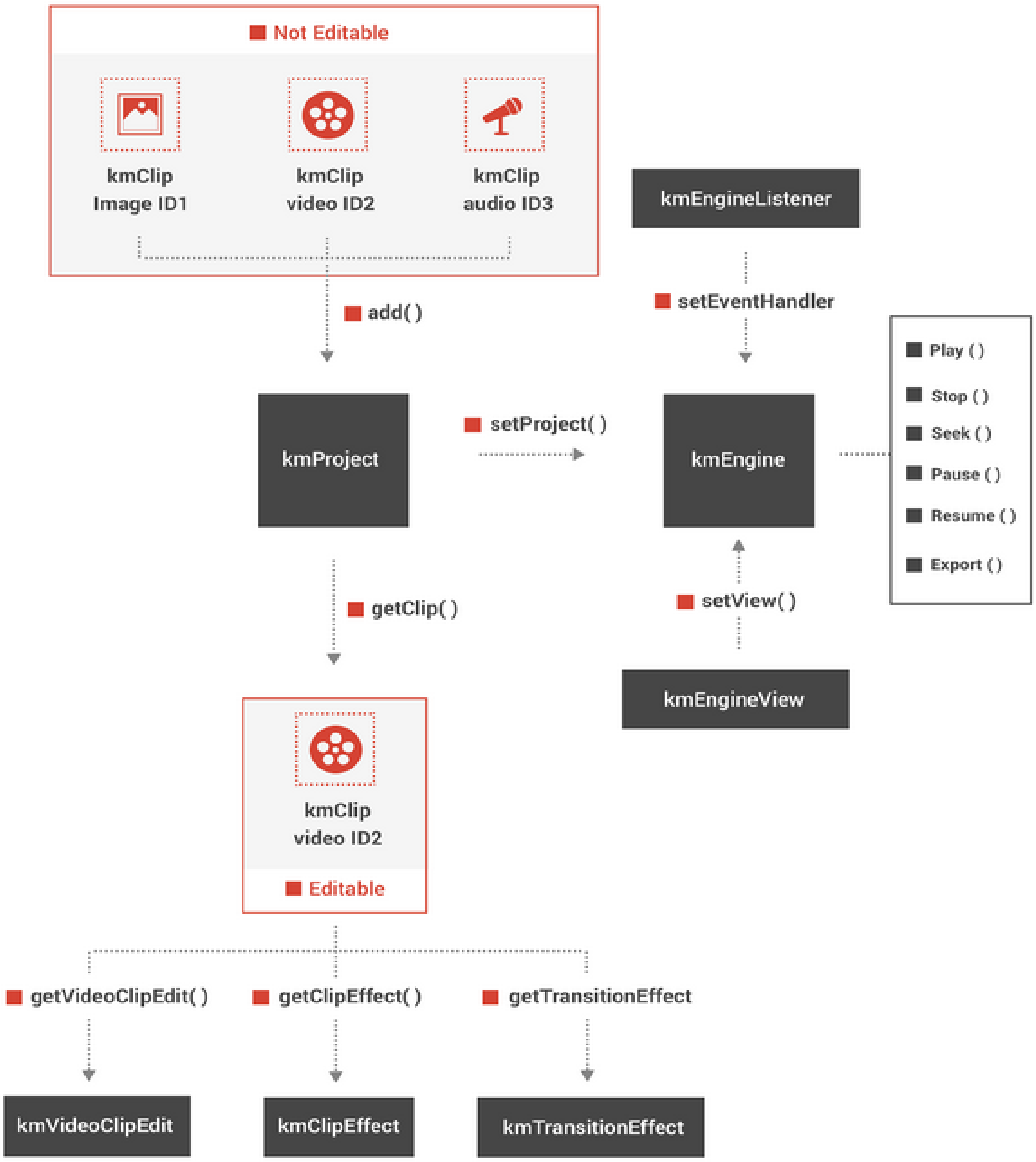


Figure 1.32: NexEditor SDK FlowChart

In the NexEditor™ SDK, all the effect modules are managed by the `nexEffectLibrary`. All the effects supported by the `nexEffectLibrary` can be added as in the flow chart diagram shown above.

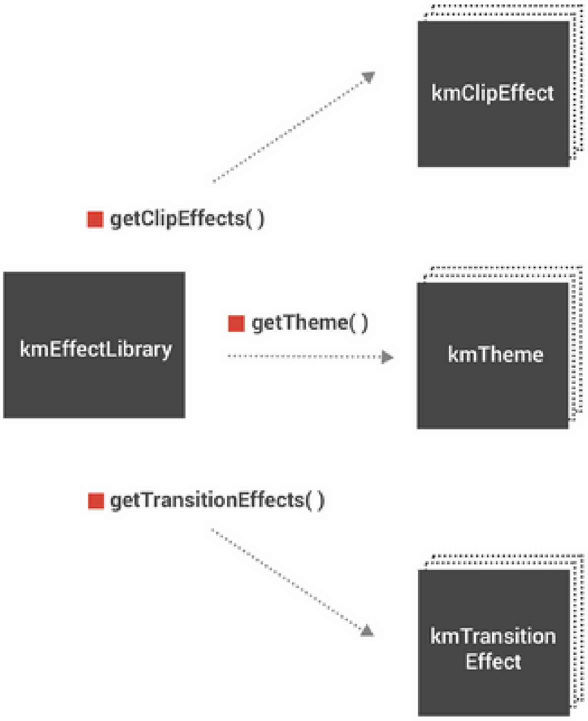


Figure 1.33: NexEditor SDK FlowChart 2

1.7 How to use NexEditor™ SDK

1.7.1 NexEditor™ SDK initialization using Application Class

NexEditor™ SDK is developed based on the singleton design pattern. Therefore, when starting the application that inherits the Android Application class, `nexApplicationConfig.createApp()` and `nexApplicationConfig.init()` must be called once.

The process above works asynchronously though so the actual application process should wait until the activity that calls `init()` has completely loaded. Use the API `nexApplicationConfig.waitForLoading( Context context, final Runnable listener );` to know when `init()` has completed loading. When `init()` has completed, `run` will be called.

Example Codes

nexApplication.java:

```
import android.app.Application;
     import android.content.Context;

     import com.nexstreaming.nexeditorsdk.nexApplicationConfig;
     import com.nexstreaming.nexeditorsdk.nexEngine;

public class nexApplication extends Application {

 private static nexEngine mEngine;
 private static Context mContext;

 public synchronized static nexEngine getEngine() {
     if(mEngine == null) {
         mEngine = new nexEngine(mContext);
     }
```

```
        return mEngine;
    }

    @Override
    public void onCreate() {
        super.onCreate();
        mContext = getApplicationContext();
        nexApplicationConfig.createApp(mContext);
        nexApplicationConfig.init(mContext, "app");
    }
}
```

AndroidManifest.xml:

```
 <?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.nexstreaming.editorsdkdemo" >

    <application
        android:name="nexApplication"
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name" >
        <activity
            ...
```

MainActivity.java:

```
public class MainActivity extends ListActivity {
   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_main);

       nexApplicationConfig.waitForLoading(this, new Runnable() {
           @Override
           public void run() {
               nexEffectLibrary fxlib = nexEffectLibrary.getEffectLibrary(MainActivity.this);
               ........
       });
   }
```

1.7.2 Loading Theme Library

Get IDs of themes, clip effects (that are not included in any existing theme), and transition effects (that are not included in any existing theme) when the theme library is completely loaded.

1. Get the effect library.
2. Get the list of theme IDs.
3. Get the list of theme exception clip effect IDs.
4. Get the list of theme exception transition effect IDs.

Example code:

```
nexEffectLibrary fxlib = nexEffectLibrary.getEffectLibrary(this);
      nexTheme[] allThemes = fxlib.getThemes();
      for( nexTheme t: allThemes ) {
          Log.i("Theme","id="+t.getId()+", name="+t.getName(getBaseContext()));
      }
```

1.7.3 Preview

Follow the steps below to preview a project (made automatically by adding clips and setting a theme) :

1. Create clips > create project > set theme IDs on the project.
2. When clips are created successfully, add them into the project.
3. Call the engine.
4. Register a listener, engine view, and the project with the engine.
5. Play the project from the engine.

Example code:

```
nexClip clip1 = nexClip.getSupportedClip("/sdcard/01.jpg");
nexClip clip2 = nexClip.getSupportedClip("/sdcard/02.mp4");
nexProject project = new nexProject(mThemeID,"Open my Create","End my Close");
project.add(clip1);
project.add(clip2);
mEngine = KMSDKApplication.getApplicationInstance().getEngine();

m_editorView = (nexEngineView)findViewById(R.id.videoView);
m_editorView.setBlackOut(true);
mEngine.setView(m_editorView);
mEngine.setProject(project);

mEngine.play();
```

1.7.4 Changing Theme

When a preview (1.7.3) was successful, to change the theme :

1. Set new theme IDs to the project.
2. Play the project from the engine.

Example code:

```
mEngin.getProject().setThemeId("com.nexstreaming.kinemaster.onstage");
mEngin.play();
```

Pre-installed Themes List

Theme name	Theme ID
Basic	com.nexstreaming.kinemaster.basic
Travel	com.nexstreaming.kinemaster.travel
On-Stage	com.nexstreaming.kinemaster.onstage
News	com.nexstreaming.kinemaster.news
70s	com.nexstreaming.kinemaster.70s
Serene	com.nexstreaming.kinemaster.serene

1.7.5 Changing Transition Effect

To change a transition effect:

1. Get a transition effect from the clip of a project, then input a new transition effect ID.
2. Play the project from the engine.

Example code:

```
mEngin.getProject().getClip(0,true).getTransitionEffect().setTransitionEffect("com.nexstreaming.kinemaster.
      builtin.transition.checker");
mEngin.play();
```

Pre-installed Transition Effect List

Transition Effect Name	Transition Effect ID
Split	com.nexstreaming.kinemaster.builtin4.split
Circle Wipe	com.nexstreaming.kinemaster.builtin2.transition.-circlewipe
Knock Aside	com.nexstreaming.kinemaster.builtin3.knockaside
Color Tiles	com.nexstreaming.kinemaster.builtin.transition.pip.-colortiles
Spin	com.nexstreaming.kinemaster.builtin4.spin
Cover	com.nexstreaming.kinemaster.builtin4.cover
Wipe	com.nexstreaming.kinemaster.builtin4.wipe
Slide	com.nexstreaming.kinemaster.builtin4.slide
Square Title	com.nexstreaming.kinemaster.builtin3.boxtexttrans
Checker Flip	com.nexstreaming.kinemaster.builtin.transition.-checker
Strip Wipe	com.nexstreaming.kinemaster.builtin4.stripwipe
Captioned Inset	com.nexstreaming.kinemaster.builtin3.-captioninsettrans
Zoom out, then in	com.nexstreaming.kinemaster.builtin.transition.-zoominout
Block in	com.nexstreaming.kinemaster.builtin4.blockin
Strips	com.nexstreaming.kinemaster.builtin.transition.strips
Many Circles	com.nexstreaming.kinemaster.builtin2.transition.-manycircles
Zoom out	com.nexstreaming.kinemaster.builtin.transition.-zoomout
Star Wipe	com.nexstreaming.kinemaster.builtin4.starwipe
Heart Wipe	com.nexstreaming.kinemaster.builtin4.heartwipe
Split-screen	com.nexstreaming.kinemaster.builtin.transition.pip.-splitscn
Clock Wipe	com.nexstreaming.kinemaster.builtin4.clockwipe
3D Zoom Flip	com.nexstreaming.kinemaster.builtin.transition.-zoomflip
Fade Through Color	com.nexstreaming.kinemaster.builtin.transition.-fadethroughcolor
Video Tiles	com.nexstreaming.kinemaster.builtin.transition.pip.-tiles

Corners	com.nexstreaming.kinemaster.builtin.transition.pip.corners
Uncover	com.nexstreaming.kinemaster.builtin4.uncover
Retro Pastel Title	com.nexstreaming.kinemaster.builtin3.retropastel
Bouncy Box	com.nexstreaming.kinemaster.builtin3.bouncebox
Inset Video	com.nexstreaming.kinemaster.builtin.transition.pip.inset
Double Split	com.nexstreaming.kinemaster.builtin4.doublesplit
Crossfade	com.nexstreaming.kinemaster.builtin.transition.crossfade
3D Flip	com.nexstreaming.kinemaster.builtin.transition.flip

1.7.6 Adding Text

To add text to a clip:

1. Choose a clip effect with text from the clip in the project, then input text.
2. Play the project from the engine.

Example code:

```
mEngine.getProject().getClip(0,true).getClipEffect().setEffect("
      com.nexstreaming.kinemaster.builtin.title.bam");
mEngin.getProject().getClip(0,true).getClipEffect().setTitle(“clip 1 title”);
mEngin.play();
```

Pre-installed Text Effect List

Text Effect Name	Text Effect ID
Subtitle	com.nexstreaming.kinemaster.builtin.title.simple
Thriller	com.nexstreaming.kinemaster.builtin.title.thriller
Bam!	com.nexstreaming.kinemaster.builtin.title.bam
Ghost Story	com.nexstreaming.kinemaster.builtin.title.ghoststory
Modern	com.nexstreaming.kinemaster.builtin.title.modern
Manuscript	com.nexstreaming.kinemaster.builtin.title.manuscript
Poem	com.nexstreaming.kinemaster.builtin6.title.poem
Flip Title	com.nexstreaming.kinemaster.builtin.title.centerflip
Stylish Subtitle	com.nexstreaming.kinemaster.builtin.title.stylishsub
Comic Book	com.nexstreaming.kinemaster.builtin.title.comic
Floating Braces	com.nexstreaming.kinemaster.builtin6.title.brackettitle
Center	com.nexstreaming.kinemaster.builtin.title.center
Elegant	com.nexstreaming.kinemaster.builtin.title.elegant
Center Low	com.nexstreaming.kinemaster.builtin.title.centerlow
Bulletin	com.nexstreaming.kinemaster.builtin.title.bulletin
Slide Across	com.nexstreaming.kinemaster.builtin5.slideacrosstxt
Stickers	com.nexstreaming.kinemaster.builtin3.overlay.sticker
Stylish Finale	com.nexstreaming.kinemaster.builtin.title.stylishfinale
Handwriting	com.nexstreaming.kinemaster.builtin.overlay.handwriting

Credits	com.nexstreaming.kinemaster.builtin.title.creditsmid
Stylish Main Title	com.nexstreaming.kinemaster.builtin.title.stylishmain

1.7.7 Speed Control

To adjust the speed of a clip:

1. Get video clip edit from the clip of the project, then input the new speed value to use. The value range of `setSpeedControl` is 25∼200.

Example code:

```
mEngin.getProject().getClip(0,true).getVideoClipEdit().setSpeedControl(200);
mEngin.play();
```

1.7.8 Export

When a preview was successful, to export the project:

1. Stop the engine.
2. Export the engine.

Example code:

```
mEngine.stop();

String sdCardPath = Environment.getExternalStorageDirectory().getAbsolutePath();
File exportDir = new File(sdCardPath + File.separator + "KM" + File.separator + "Export");
exportDir.mkdirs();
File exportFile = new File(exportDir, "NexEditor.mp4");

mEngine.export(exportFile .getAbsolutePath(), 1280, 720, 6 * 1024 * 1024, Long.MAX_VALUE);
```

1.7.9 Adding Theme Files

Follow the steps below to add theme (KMT) files to the SD card and to the application.

Add KMT to the SD card :

1. Create **NexDemo** > **Plugin** directory.
2. Copy compressed KMT file into the Plugin folder.
3. The resource files for the themes should be saved into the **Plugin** > **externals** folder.

Add KMT to the Application Asset :

1. Create **App** > **assets** > **km** directory.

2. Add KMT folder, uncompressed, into the **App** > **assets** > **km** folder.
3. Add **.kmt** at the end of the KMT folder name. (App > assets > km > KMT.kmt)
4. The resource files for the KMT (.mp4/.aac/.jpg) should be saved into **assets** > **km** > **shared** directory.

Author: NexStreaming Corp.

Chapter 2

Deprecated List

Member com.nexstreaming.nexeditorsdk.nexClip.cancelThumbnails ()

For internal use only. Please do not use.

Member com.nexstreaming.nexeditorsdk.nexClip.getVideoClipIDR2YOnlyThumbnails (int thumbWidth, int thumbHeight, final int startTime, final int endTime, int maxCount, final OnGetVideoClipIDR2YOnlyThumbnailsListener listener)

For internal use only. Please do not use.

Member com.nexstreaming.nexeditorsdk.nexClip.getVideoClipIFrameThumbnails (int thumbWidth, int thumbHeight, final OnGetVideoClipDetailThumbnailsListener listener)

For internal use only. Please do not use.

Class com.nexstreaming.nexeditorsdk.nexClip.OnGetVideoClipIDR2YOnlyThumbnailsListener

For internal use only. Please do not use.

Since

version 1.3.43

Member com.nexstreaming.nexeditorsdk.nexClip.setThumbTempDir (String path)

For internal use only. Please do not use.

Member com.nexstreaming.nexeditorsdk.nexClip.setVideoClipDetailThumbnailsDiskLimit (long size)

For internal use only. Please do not use.

Member com.nexstreaming.nexeditorsdk.nexEngine.cancelKineMixExport ()

For internal use only. Please do not use.

Member com.nexstreaming.nexeditorsdk.nexEngine.checkKineMixExport (boolean dummy)

For internal use only. Please do not use.

Member com.nexstreaming.nexeditorsdk.nexEngine.checkKineMixExport ()

For internal use only. Please do not use.

Member com.nexstreaming.nexeditorsdk.nexEngine.checkKineMixExport (String srcFilePath, String dstFilePath)

For internal use only. Please do not use.

Member com.nexstreaming.nexeditorsdk.nexEngine.KineMixExport (String strExportFile)

For internal use only. Please do not use.

Member com.nexstreaming.nexeditorsdk.nexEngine.stopSync ()

For internal use only. Please do not use.

Member com.nexstreaming.nexeditorsdk.nexEngineListener.onClipInfoDone ()

For internal use only. Please do not use.

Member com.nexstreaming.nexeditorsdk.nexEngineView.setBlackOut (boolean blackOut)

For internal use only. Please do not use.

Member com.nexstreaming.nexeditorsdk.nexOverlayItem.getLayerExpression ()

This method gets the index of a preset animation effect. Use `getLayerExpressions()` to get the label names.

For internal use only. Please do not use.

Member com.nexstreaming.nexeditorsdk.nexOverlayItem.getLayerExpressionDuration ()

This method gets the duration of a preset animation effect for a clip.

Returns

Member com.nexstreaming.nexeditorsdk.nexOverlayItem.getLayerExpressiontParam ()

Member com.nexstreaming.nexeditorsdk.nexOverlayItem.setLayerExpression (nexOverlayKineMasterExpression layerExpressionIndex)

This method sets an effect for a preset animation. If the parameter `layerExpression` is set to 0, an animation effect made by the developer should be used instead of a preset animation effect.

Member com.nexstreaming.nexeditorsdk.nexOverlayItem.setLayerExpressionDuration (int duration)

This method sets the duration of a preset animation effect for a clip.

Member com.nexstreaming.nexeditorsdk.nexProject.getSecondaryItems ()

For internal use only. Please do not use.

Member com.nexstreaming.nexeditorsdk.nexProject.getTotalVisualClipCount ()

For internal use only. Please do not use.

Member com.nexstreaming.nexeditorsdk.nexProject.makeKineMasterIntent ()

For internal use only. Please do not use.

Member com.nexstreaming.nexeditorsdk.nexVideoClipEdit.addTrim (int startTime, int endTime, int speed)

This method can be used to trim several parts of a video clip in different ways.

For internal use only. Please do not use.

Member com.nexstreaming.nexeditorsdk.nexVideoClipEdit.getTrimCount ()

This method gets the number of times trim has been set with the method addTrim() within a specific clip.

Returns

The number of trims set for the specific clip.

Since

version 1.0.0

See Also

#addTrim(int, int, int)
#removeTrim(int)

For internal use only. Please do not use.

Member com.nexstreaming.nexeditorsdk.nexVideoClipEdit.kAutoTrim_Divided

For internal use only. Please do not use.

Member com.nexstreaming.nexeditorsdk.nexVideoClipEdit.kAutoTrim_Interval

For internal use only. Please do not use.

Member com.nexstreaming.nexeditorsdk.nexVideoClipEdit.removeTrim (int loc)

This method removes a trim set on a video clip.

Parameters

loc	The location of the trim to be removed.

Since

version 1.0.0

See Also

#addTrim(int, int, int)
#setTrim(int, int)

For internal use only. Please do not use.

Member com.nexstreaming.nexeditorsdk.nexVideoClipEdit.setAutoTrim (int trimMode, int value)

This method trims a video clip automatically, based on the mode chosen with the parameter, `trimMode`.

Parameters

trimMode	How to automatically trim video clips. This should be one of:

For internal use only. Please do not use.

Chapter 3

Namespace Documentation

3.1 Package com.nexstreaming.nexeditorsdk

Classes

- class nexAnimate
- class nexApplicationConfig
- class nexAssetMediaManager
- class nexAssetPackageManager

  nexAssetPackageManager is a class that installs and manages any downloaded Assets from the Asset Store Application from NexStreaming.
- class nexAssetStoreAppUtils
- class nexAudioEdit
- class nexAudioEnvelop
- class nexAudioItem
- class nexChecker
- class nexClip
- class nexClipEffect
- class nexColorEffect
- class nexConfig
- class nexCrop
- class nexEffect
- class nexEffectLibrary
- class nexEffectOptions
- class nexEffectPreviewView
- class nexEngine
- interface nexEngineListener
- class nexEngineView
- class nexFont
- class **nexObserver**
- class nexOverlayImage
- class nexOverlayItem
- class nexOverlayKineMasterExpression
- class nexOverlayKineMasterText
- class nexOverlayManager
- class nexOverlayPreset
- class nexProject
- class nexTemplate
- class nexTemplateComposer
- class nexTemplateManager

This class provides an easy way to use items of the template Category in nexAssetPackageManager in nexTemplateManager. If the developer uses only templates of nexEditorSDK or separates templates, they can choose not to use nexAssetPackageManager and compose an application with just nexTemplateManager.

- class nexTemplateTitle
- class nexTheme
- class nexTranscode
- class nexTransitionEffect
- class nexUtils
- class nexVideoClipEdit

3.1.1 Detailed Description

NexEditor™ SDK Package

Desc :

Version

version 1.0.0

Author

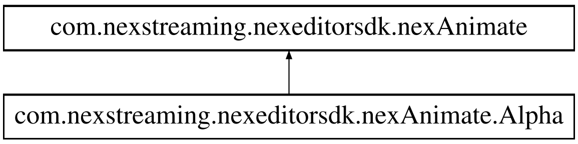
Nexstreaming Corp.

Chapter 4

Class Documentation

4.1 com.nexstreaming.nexeditorsdk.nexAnimate.Alpha Class Reference

Inheritance diagram for com.nexstreaming.nexeditorsdk.nexAnimate.Alpha:


com.nexstreaming.nexeditorsdk.nexAnimate
com.nexstreaming.nexeditorsdk.nexAnimate.Alpha

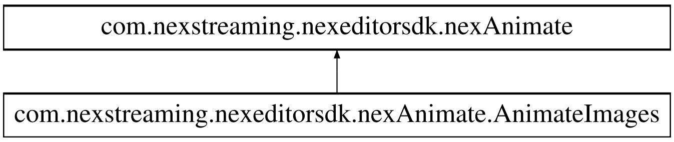
Protected Member Functions

- **Alpha** (int startTime, int duration, float start, float end)
- float **getAlpha** (int time)

Additional Inherited Members

4.2 com.nexstreaming.nexeditorsdk.nexAnimate.AnimateImages Class Reference

Inheritance diagram for com.nexstreaming.nexeditorsdk.nexAnimate.AnimateImages:


com.nexstreaming.nexeditorsdk.nexAnimate
com.nexstreaming.nexeditorsdk.nexAnimate.AnimateImages

Protected Member Functions

- **AnimateImages** (int startTime, int duration, int...ids)
- int **getImageResourceId** (int currTime)

Additional Inherited Members

4.3 com.nexstreaming.nexeditorsdk.nexApplicationConfig.APILevel Enum Reference

Public Member Functions

- **APILevel** (int val)
- int **getValue** ()

Static Public Member Functions

- static APILevel **fromValue** (int value)

Public Attributes

- **UnLimited** =(EditorGlobal.kUnLimited)
- **OverlayImageLimited** =(EditorGlobal.kOverlayImageLimited)
- **OverlayVideoLimited** =(EditorGlobal.kOverlayVideoLimited)
- **OverlayAnimateLimited** =(EditorGlobal.kOverlayAnimateLimited)

4.3.1 Detailed Description

This enumeration defines the enumeration types of features provided by the NexEditor™SDK.

These enumeration types include:

- **UnLimited**: All features are available to use.
- **OverlayImageLimited**: OverlayImage, OverlayVideo, OverlayAnimate features can not be used.
- **OverlayVideoLimited**: OverlayVideo, OverlayAnimate features can not be used.
- **OverlayAnimateLimited**: OverlayAnimate feature can not be used.

Since

version 1.5.19

4.4 com.nexstreaming.nexeditorsdk.nexAssetPackageManager.Asset Interface Reference

Inherited by com.nexstreaming.nexeditorsdk.nexAssetPackageManager.AssetEnt.

Public Member Functions

- int assetIdx ()
- String assetId ()
- String assetName (String locale)
- String priceType ()
- long installedTime ()
- long expireRemain ()
- AssetInstallType installedType ()
- String getCategoryAlias ()

4.4.1 Detailed Description

The Asset Package information.

Since

1.7.0

4.4.2 Member Function Documentation

4.4.2.1 String com.nexstreaming.nexeditorsdk.nexAssetPackageManager.Asset.assetId ( )

The Asset ID.

If it is App_Asset or Share type, the rootDirectory name of the package.

Returns

Since

1.7.0

4.4.2.2 int com.nexstreaming.nexeditorsdk.nexAssetPackageManager.Asset.assetIdx ( )

The Asset Store index.

If it is App_Asset or Share type, the value has no meaning.

Returns

Since

1.7.0

4.4.2.3 String com.nexstreaming.nexeditorsdk.nexAssetPackageManager.Asset.assetName (String *locale*)

The Asset name. The developer must check if it is `NULL`. (null in some cases)

Parameters

locale	

Returns

Since

1.7.0

4.4.2.4 long com.nexstreaming.nexeditorsdk.nexAssetPackageManager.Asset.expireRemain ( )

Preparing...

Returns

Since

1.7.0

4.4.2.5 String com.nexstreaming.nexeditorsdk.nexAssetPackageManager.Asset.getCategoryAlias ( )

Asset category alias name.

Returns

Since

1.7.1

4.4.2.6 long com.nexstreaming.nexeditorsdk.nexAssetPackageManager.Asset.installedTime ( )

Preparing...

Returns

Since

1.7.0

4.4.2.7 AssetInstallType com.nexstreaming.nexeditorsdk.nexAssetPackageManager.Asset.installedType ( )

This gets the installation type of the Asset.

Returns

STORE - Downloaded from the Asset Store, SHARE - Dev shared folder, APP_ASSETS - Assets folder of the App.

Since

1.7.0

4.4.2.8 String com.nexstreaming.nexeditorsdk.nexAssetPackageManager.Asset.priceType ( )

Preparing...

Returns

Since

1.7.0

4.5 com.nexstreaming.nexeditorsdk.nexAssetPackageManager.AssetInstallType Enum Reference

Public Attributes

- **STORE**
- **SHARE**
- **APP_ASSETS**

4.5.1 Detailed Description

The value of Asset#installedType().

See Also

Asset#installedType()

Since

1.7.0

4.6 com.nexstreaming.nexeditorsdk.nexAssetMediaManager.AssetMedia Class Reference

Inherits com.nexstreaming.nexeditorsdk.nexAssetPackageManager.ItemEnt.

Public Member Functions

- String getPath ()
- int getClipType ()
- boolean filter ()
- Bitmap thumbnail ()

4.6.1 Detailed Description

This is created by inheriting com.nexstreaming.nexeditorsdk.nexAssetPackageManager.Item, which is a media class of an Asset Item.

Since

1.7.1

4.6.2 Member Function Documentation

4.6.2.1 boolean com.nexstreaming.nexeditorsdk.nexAssetMediaManager.AssetMedia.filter ( )

Assets may include video content that processes sound effects or background images from different media. Media like this has a property of Filter.

Returns

`TRUE` - media for filters.

4.6.2.2 int com.nexstreaming.nexeditorsdk.nexAssetMediaManager.AssetMedia.getClipType ( )

This gets the AssetMedia Type value, which is the same as that of nexClip#getClipType().

Returns

nexClip.kCLIP_TYPE_AUDIO - audio, nexClip.kCLIP_TYPE_VIDEO - video, nexClip.kCLIP_TYPE_IMAGE - image

See Also

nexClip::getClipType()

Since

1.7.1

4.6.2.3 String com.nexstreaming.nexeditorsdk.nexAssetMediaManager.AssetMedia.getPath ( )

This gets the Media Path. For details, please see nexAssetPackageManager#getAssetPackageMediaPath(Context, String).

Returns

The Absolute Path of content.

See Also

nexAssetPackageManager::getAssetPackageMediaPath(Context, String)

Since

1.7.1

4.6.2.4 Bitmap com.nexstreaming.nexeditorsdk.nexAssetMediaManager.AssetMedia.thumbnail ( )

If the type is nexClip#kCLIP_TYPE_IMAGE, then the developer can get the image thumbnail.

Returns

`NULL` - No thumbnail or incorrect ID OR it is not nexClip.kCLIP_TYPE_IMAGE

Since

1.7.1

4.7 com.nexstreaming.nexeditorsdk.nexOverlayItem.BoundInfo Class Reference

Public Member Functions

- float getScaleX ()
- float getScaleY ()
- float getTranslateX ()
- float getTranslateY ()
- float getAngle ()
- int getWidth ()
- int getHeight ()
- int getTime ()

4.7.1 Detailed Description

This class indicates the position and state of the overlay item in the coordinates.

Since

version 1.5.3

4.7.2 Member Function Documentation

4.7.2.1 float com.nexstreaming.nexeditorsdk.nexOverlayItem.BoundInfo.getAngle ( )

This method gets the rotation value of the overlay item as of now.

Returns

The rotation value of the overlay item at a set time.

Since

version 1.5.3

4.7.2.2 int com.nexstreaming.nexeditorsdk.nexOverlayItem.BoundInfo.getHeight ( )

This method gets the height value of the overlay item as of now.

Returns

The height value of the overlay item at a set time.

Since

version 1.5.3

4.7.2.3 float com.nexstreaming.nexeditorsdk.nexOverlayItem.BoundInfo.getScaleX ( )

This method gets the x-scale value of the overlay item as of now.

Returns

The x scale value of the overlay item at a set time.

Since

version 1.5.3

4.7.2.4 float com.nexstreaming.nexeditorsdk.nexOverlayItem.BoundInfo.getScaleY ( )

This method gets the y-scale value of the overlay item as of now.

Returns

The y scale value of the overlay item at a set time.

Since

version 1.5.3

4.7.2.5 int com.nexstreaming.nexeditorsdk.nexOverlayItem.BoundInfo.getTime ( )

This method gets the time value of the project timeline as of now.

Returns

The set time.

Since

version 1.5.3

4.7.2.6 float com.nexstreaming.nexeditorsdk.nexOverlayItem.BoundInfo.getTranslateX ( )

This method gets the x-axis value of the overlay item as of now.

Returns

The x-axis value of the overlay item at a set time.

Since

version 1.5.3

4.7.2.7 float com.nexstreaming.nexeditorsdk.nexOverlayItem.BoundInfo.getTranslateY ( )

This method gets the y-axis value of the overlay item as of now.

Returns

The y-axis value of the overlay item at a set time.

Since

version 1.5.3

4.7.2.8 int com.nexstreaming.nexeditorsdk.nexOverlayItem.BoundInfo.getWidth ( )

This method gets the width value of the overlay item as of now.

Returns

The width value of the overlay item at a set time.

Since

version 1.5.3

4.8 com.nexstreaming.nexeditorsdk.nexAssetPackageManager.Category Enum Reference

Public Attributes

- **audio**
- **audiofilter**
- **background**
- **effect**
- **filter**
- **font**
- **overlay**
- **template**
- **transition**
- **extra**

4.8.1 Detailed Description

The value of Item#category().

See Also

Item#category()

Since

1.7.0

4.9 com.nexstreaming.nexeditorsdk.nexOverlayItem.ChromaKey Class Reference

Public Member Functions

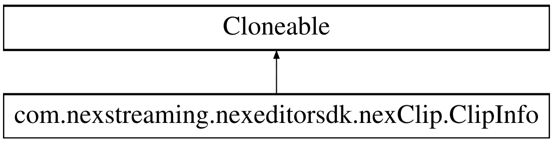
- void **getChromaKeyDivisions** (float[ ] divisions)
- void **getChromaKeyStrengths** (float[ ] strengths)
- void **setChromaKeyStrengths** (float[ ] strengths)
- int **getChromaKeyColor** ()
- void **setChromaKeyColor** (int color)
- boolean **getChromaKeyEnabled** ()
- void **setChromaKeyMaskEnabled** (boolean enabled)
- boolean **getChromaKeyMaskEnabled** ()
- void **setChromaKeyEnabled** (boolean enabled)
- void **setChromaKeyFGClip** (float fgClip)
- void **setChromaKeyBGClip** (float bgClip)
- float **getChromaKeyFGClip** ()
- float **getChromaKeyBGClip** ()
- void **setChromaKeyBlend** (float[ ] xyxy)
- void **getChromaKeyBlend** (float[ ] xyxy)

Static Public Member Functions

- static int[ ] **getChromaKeyRecommendedColors** (Bitmap thumb)

4.10 com.nexstreaming.nexeditorsdk.nexClip.ClipInfo Class Reference

Inheritance diagram for com.nexstreaming.nexeditorsdk.nexClip.ClipInfo:



Public Attributes

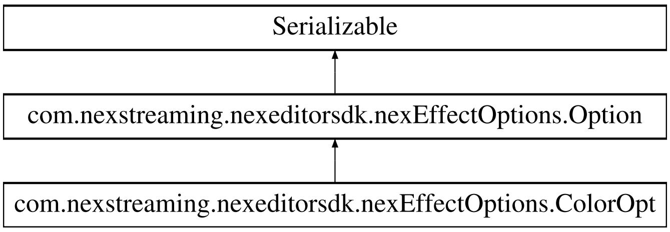
- String **mMimeType**
- int **mClipType** = kCLIP_TYPE_NONE
- int **mWidth**
- int **mHeight**
- int **mDisplayWidth**
- int **mDisplayHeight**
- boolean **mExistVideo**
- boolean **mExistAudio**
- int **mTotalTime**
- int **mFramesPerSecond**
- int **mRotateDegreeInMeta**
- int **mH264Profile**
- int **mH264Level**
- int **mSuppertedResult** = kClip_Supported_Unknown
- int **mVideoBitrate**
- int **mAudioBitrate**
- int **mVideoTotalTime**
- int **mAudioTotalTime**
- int **mSeekPointCount**
- int **mVideoRenderMode**
- byte[ ] **mVideoUUID**

Static Protected Member Functions

- static ClipInfo **clone** (ClipInfo src)

4.11 com.nexstreaming.nexeditorsdk.nexEffectOptions.ColorOpt Class Reference

Inheritance diagram for com.nexstreaming.nexeditorsdk.nexEffectOptions.ColorOpt:


Serializable
com.nexstreaming.nexeditorsdk.nexEffectOptions.Option
com.nexstreaming.nexeditorsdk.nexEffectOptions.ColorOpt

Public Member Functions

- int getARGBformat ()
- void setARGBColor (int ARGBformat)

Protected Member Functions

- **ColorOpt** (String id, String label, String color)

Protected Attributes

- int **default_argb_color**
- int **argb_color**

4.11.1 Detailed Description

This class saves the possible color-related options for an effect.

Possible color-related options for an effect may include text or background colors among others, depending on the specific effect.

Example code:

```
for( nexEffectOptions.ColorOpt opt : color){
            if( f.getId().compareTo(opt.getId()) == 0 ){
                b.append(URLEncoder.encode(opt.getId(), "UTF-8"));
                b.append("=");
                b.append(URLEncoder.encode(ColorUtil.colorString(opt.getARGBformat()) ,"UTF-8"));
            }
        }
```

See Also

getColorOptions()

Since

version 1.0.1

4.11.2 Member Function Documentation

4.11.2.1 int com.nexstreaming.nexeditorsdk.nexEffectOptions.ColorOpt.getARGBformat ( )

This method gets the saved color for a color-related option, as an ARGB 32-bit value.

Returns

The option color in ARGB format, as an `integer`.

Since

version 1.0.1

4.11.2.2 void com.nexstreaming.nexeditorsdk.nexEffectOptions.ColorOpt.setARGBColor (int *ARGBformat*)

This method sets the color for a color-related option as an ARGB 32-bit value.

Parameters

ARGBformat	The color to set in ARGB format, as an `integer`.

Since

version 1.0.1

4.12 com.nexstreaming.nexeditorsdk.nexCrop.CropMode Enum Reference

Public Attributes

- **FIT**
- **FILL**
- **PAN_RAND**
- **PAN_FACE**

4.12.1 Detailed Description

This enumeration defines the possible `CropMode` options when a clip is cropped.

These options are possible values for the `cropMode` parameter of the `randomizeStartEndPosition()` method.

The possible modes for a crop are:

- **FIT:** The largest dimension of the original clip will be fit to the project display, without any distortion of the original dimensions (and the rest of the display will display black).

- **FILL:** The clip will be cropped to fill the project display, and the original may be distorted.
- **PAN_RAND:** The start and end positions of the crop will be randomly selected.
- **PAN_FACE:** Not currently available.

See Also

randomizeStartEndPosition(boolean, com.nexstreaming.nexeditorsdk.nexCrop.CropMode)

Since

version 1.1.0

4.13 com.nexstreaming.nexeditorsdk.nexTranscode.Error Enum Reference

Public Attributes

- **NONE**
- **NOTSUPPORTEDFILE**
- **ENGINEFAIL**
- **SOURCEFAIL**
- **BUSY**
- **RUNFAIL**
- **CANCEL**

4.13.1 Detailed Description

The error value of nexTranscode.

See Also

getLastError()

Since

version 1.5.29

4.14 com.nexstreaming.nexeditorsdk.nexEngine.FastPreviewOption Enum Reference

Public Attributes

- **normal**
- **brightness**
- **contrast**
- **saturation**
- **tintColor**
- **cts**

4.14.1 Detailed Description

This enumeration defines the fast preview options to be used by the method `fastPreview()`.

The preview options include:

- **normal**,
- **brightness**,
- **contrast**, and
- **saturation**.

  Note

  `Normal` effect is not used in the current version.

  See Also

  fastPreview(FastPreviewOption, int)

  Since

  version 1.1.0

4.15 com.nexstreaming.nexeditorsdk.nexOverlayItem.HitPoint Class Reference

Public Member Functions

- int getID ()
- int getHitInPosition ()

Public Attributes

- int mTime
- float mViewX
- float mViewY
- float mViewWidth
- float mViewHeight

4.15.1 Detailed Description

This method gets the information of touched location on the screen. The information includes X and Y coordinates, width and height of the view.

Example code :

```
m_editorView = (nexEngineView)findViewById(R.id.engineview_overlay_fastview);
    m_editorView.setOnTouchListener(new View.OnTouchListener() {
        nexOverlayItem.HitPoint mPosition = new nexOverlayItem.HitPoint();
        mPosition.mTime = 0;
        mPosition.mViewX = event.getX();
        mPosition.mViewY = event.getY();
        mPosition.mViewWidth = v.getWidth();
        mPosition.mViewHeight = v.getHeight();
        //
    }
```

Since

version 1.5.3

4.15.2 Member Function Documentation

4.15.2.1 int com.nexstreaming.nexeditorsdk.nexOverlayItem.HitPoint.getHitInPosition ( )

This method gets the hit position of this overlay.

Returns

Zero for when the hit position is in the center area; otherwise, a non-zero value.

- kOutLine_Pos_LeftTop={ kOutLine_Pos_LeftTop} , kOutLine_Pos_RightTop={ kOutLine_Pos_RightTop} , kOutLine_Pos_LeftBottom={ kOutLine_Pos_LeftBottom} ,kOutLine_Pos_RightBottom={ kOutLine_Pos_RightBottom}

Since

version 1.5.3

4.15.2.2 int com.nexstreaming.nexeditorsdk.nexOverlayItem.HitPoint.getID ( )

This method gets the ID of the selected overlay item.

Returns

Zero for when there is no selected overlay items, or a non-zero ID value for the selected overlay item.

Since

version 1.5.3

4.15.3 Member Data Documentation

4.15.3.1 int com.nexstreaming.nexeditorsdk.nexOverlayItem.HitPoint.mTime

This method gets the selected time from the overlay timeline.

Warning

The time inside the set duration of this overlay, not the project timeline.

Since

version 1.5.3

4.15.3.2 float com.nexstreaming.nexeditorsdk.nexOverlayItem.HitPoint.mViewHeight

This method gets the view height value entered in the overlay coordinates.

Since

version 1.5.3

4.15.3.3 float com.nexstreaming.nexeditorsdk.nexOverlayItem.HitPoint.mViewWidth

This method gets the view width value entered in the overlay coordinates.

Since

version 1.5.3

4.15.3.4 float com.nexstreaming.nexeditorsdk.nexOverlayItem.HitPoint.mViewX

This method gets the view x value entered in the overlay coordinates.

Since

version 1.5.3

4.15.3.5 float com.nexstreaming.nexeditorsdk.nexOverlayItem.HitPoint.mViewY

This method gets the view y value entered in the overlay coordinates.

Since

version 1.5.3

4.16 com.nexstreaming.nexeditorsdk.nexAssetPackageManager.Item Interface Reference

Inherited by com.nexstreaming.nexeditorsdk.nexAssetPackageManager.ItemEnt.

Public Member Functions

- String name (String locale)
- String id ()
- Asset **packageInfo** ()
- boolean validate ()

- boolean isDelete ()
- Bitmap icon ()
- nexAssetPackageManager.Category category ()
- boolean hidden ()

4.16.1 Detailed Description

This is an interface which gets the information of an installed Asset.

Since

1.7.0

4.16.2 Member Function Documentation

4.16.2.1 nexAssetPackageManager.Category com.nexstreaming.nexeditorsdk.nexAssetPackageManager.Item.category ( )

This is the item category.

Returns

The item category.

Since

1.7.0

4.16.2.2 boolean com.nexstreaming.nexeditorsdk.nexAssetPackageManager.Item.hidden ( )

This gets the hidden property of an item. Sometimes, the hidden property is set to `TRUE` in order to prevent the end user from using items for other purposes such as editing, combining with other items, and ultimately causing license issues. In this case, the application must process it so that the end user cannot see this item. Basically, if this property is set to `TRUE`, the list of nexTemplateManager, nexFont, nexOverlayPreset, nexAssetMediaManager, and nexEffectLibrary will not appear visible.

Returns

`TRUE` - the end user will not see this item, FALSE - visible and can be used for other purposes.

Since

1.7.1

4.16.2.3 Bitmap com.nexstreaming.nexeditorsdk.nexAssetPackageManager.Item.icon ( )

This returns the icon in a package as a Bitmap.

If there is no icon, it returns a thumbnail; `NULL` if thumbnail does not exist.

Returns

`NULL` - No icon set to the item.

Since

1.7.0

4.16.2.4 String com.nexstreaming.nexeditorsdk.nexAssetPackageManager.Item.id ( )

The Asset ID.

Returns

The Asset ID.

Since

1.7.0

Implemented in com.nexstreaming.nexeditorsdk.nexTemplateManager.Template.

4.16.2.5 boolean com.nexstreaming.nexeditorsdk.nexAssetPackageManager.Item.isDelete ( )

Only downloaded Assets from the Asset Store can be deleted. This API notifies whether or not the asset can be uninstalled.

Returns

`TRUE` - Can be deleted, `FALSE` - Do not delete.

See Also

uninstallPackageById(String)
uninstallPackageByAssetIdx(int)

Since

1.7.0

4.16.2.6 String com.nexstreaming.nexeditorsdk.nexAssetPackageManager.Item.name (String *locale*)

This gets the item label. If there is no label set to the Asset, it is same as id().

Parameters

locale	The language or region information. For example, "en" - English , "ko_KR"- Korean, `NULL` is default value.

Returns

The label name.

Since

1.7.0

Implemented in com.nexstreaming.nexeditorsdk.nexTemplateManager.Template.

4.16.2.7 boolean com.nexstreaming.nexeditorsdk.nexAssetPackageManager.Item.validate ( )

Preparing...

Returns

Since

1.7.0

4.17 com.nexstreaming.nexeditorsdk.nexOverlayItem.Mask Class Reference

Public Member Functions

- void setState (boolean onOff)
- boolean getState ()
- void setAngle (int degree)
- int getAngle ()
- int getSplitMode ()
- int width ()
- int height ()
- void setSplitMode (int splitMode, boolean vertical)
- void setPosition (int left, int top, int right, int bottom)
- void setPosition (Rect rect)
- void getPosition (Rect rect)

Static Public Attributes

- static final int kSplit_Left = 1
- static final int kSplit_Right = 2
- static final int kSplit_Top = 3
- static final int kSplit_Bottom = 4

- static final int kSplit_LeftTop = 5
- static final int kSplit_RightTop = 6
- static final int kSplit_LeftBottom = 7
- static final int kSplit_RightBottom = 8

4.17.1 Detailed Description

This class allows masking parts of the overlay image.

Note

Only rectangular masks are supported in version 1.5.3.

Since

version 1.5.3

4.17.2 Member Function Documentation

4.17.2.1 int com.nexstreaming.nexeditorsdk.nexOverlayItem.Mask.getAngle ( )

This method gets the set rotation value to the setAngle(int).

Returns

The value of the angle.

Since

version 1.5.3

4.17.2.2 void com.nexstreaming.nexeditorsdk.nexOverlayItem.Mask.getPosition (Rect *rect*)

This method sets the mask position in the overlay coordinates.

Parameters

rect	The area to mask.

Since

version 1.5.3

4.17.2.3 int com.nexstreaming.nexeditorsdk.nexOverlayItem.Mask.getSplitMode ( )

This method gets the value of the SplitMode, which is a predefined mask.

Returns

The SplitMode value.

Since

version 1.5.3

4.17.2.4 boolean com.nexstreaming.nexeditorsdk.nexOverlayItem.Mask.getState ( )

This method gets the activation state of the mask.

Returns

`TRUE` if activated; otherwise, `FALSE`.

Since

version 1.5.3

4.17.2.5 int com.nexstreaming.nexeditorsdk.nexOverlayItem.Mask.height ( )

This method indicates the height of the mask.

Returns

The height of the mask.

Since

version 1.5.3

4.17.2.6 void com.nexstreaming.nexeditorsdk.nexOverlayItem.Mask.setAngle (int *degree*)

This method rotates the masked image counterclockwise.

Parameters

degree	(0 - 360)

Since

version 1.5.3

4.17.2.7 void com.nexstreaming.nexeditorsdk.nexOverlayItem.Mask.setPosition (int *left*, int *top*, int *right*, int *bottom*)

This method sets the mask position in the overlay coordinates. Used when freely setting the mask position without using the setSplitMode(int, boolean).

Parameters

left	The left part of the mask.
top	The top part of the mask.
right	The right part of the mask.
bottom	The bottom part of the mask.

Since

version 1.5.3

4.17.2.8 void com.nexstreaming.nexeditorsdk.nexOverlayItem.Mask.setPosition (Rect *rect*)

This method sets the mask position in the overlay coordinates. Used when freely setting the mask position without using the setSplitMode(int, boolean).

Parameters

rect	The area to mask.

Since

version 1.5.3

4.17.2.9 void com.nexstreaming.nexeditorsdk.nexOverlayItem.Mask.setSplitMode (int *splitMode,* boolean *vertical*)

This method sets the splitting mode for the overlay image.

Parameters

splitMode	The splitting mode defined in the class `Mask`.
vertical	Decides which way to split the overlay image. • **TRUE** Split into 2 columns. • **FALSE** Split into 2 rows.

Since

version 1.5.3

4.17.2.10 void com.nexstreaming.nexeditorsdk.nexOverlayItem.Mask.setState (boolean *onOff*)

This method sets the activation state of the mask.

Parameters

onOff	Decides whether or not to activate the mask. • **TRUE** Activate. • **FALSE** Deactivate.

Since

version 1.5.3

4.17.2.11 int com.nexstreaming.nexeditorsdk.nexOverlayItem.Mask.width ( )

This method indicates the width of the mask.

Returns

The width of the mask.

Since

version 1.5.3

4.17.3 Member Data Documentation

4.17.3.1 final int com.nexstreaming.nexeditorsdk.nexOverlayItem.Mask.kSplit_Bottom = 4 [static]

This method splits the overlay image into 2 rows and only shows the bottom, if using setSplitMode(int, boolean).

Since

version 1.5.3

4.17.3.2 final int com.nexstreaming.nexeditorsdk.nexOverlayItem.Mask.kSplit_Left = 1 [static]

This method splits the overlay image into 2 columns and only shows the left, if using setSplitMode(int, boolean).

Since

version 1.5.3

4.17.3.3 final int com.nexstreaming.nexeditorsdk.nexOverlayItem.Mask.kSplit_LeftBottom = 7 [static]

This method splits the overlay image into 2 x 2 blocks and only shows the bottom-left, if using setSplitMode(int, boolean).

Since

version 1.5.3

4.17.3.4 final int com.nexstreaming.nexeditorsdk.nexOverlayItem.Mask.kSplit_LeftTop = 5 [static]

This method splits the overlay image into 2 x 2 blocks and only shows the top-left, if using setSplitMode(int, boolean).

Since

version 1.5.3

4.17.3.5 final int com.nexstreaming.nexeditorsdk.nexOverlayItem.Mask.kSplit_Right = 2 [static]

This method splits the overlay image into 2 columns and only shows the right, if using setSplitMode(int, boolean).

Since

version 1.5.3

4.17.3.6 final int com.nexstreaming.nexeditorsdk.nexOverlayItem.Mask.kSplit_RightBottom = 8 [static]

This method splits the overlay image into 2 x 2 blocks and only shows the bottom-right, if using setSplitMode(int, boolean).

Since

version 1.5.3

4.17.3.7 final int com.nexstreaming.nexeditorsdk.nexOverlayItem.Mask.kSplit_RightTop = 6 [static]

This method splits the overlay image into 2 x 2 blocks and only shows the top-right, if using setSplitMode(int, boolean).

Since

version 1.5.3

4.17.3.8 final int com.nexstreaming.nexeditorsdk.nexOverlayItem.Mask.kSplit_Top = 3 [static]

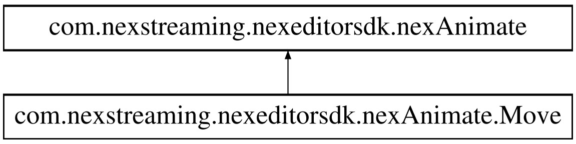
This method splits the overlay image into 2 rows and only shows the top, if using setSplitMode(int, boolean).

Since

version 1.5.3

4.18 com.nexstreaming.nexeditorsdk.nexAnimate.Move Class Reference

Inheritance diagram for com.nexstreaming.nexeditorsdk.nexAnimate.Move:


com.nexstreaming.nexeditorsdk.nexAnimate
com.nexstreaming.nexeditorsdk.nexAnimate.Move

Protected Member Functions

- **Move** (int startTime, int duration, MoveTrackingPath path)
- float **getTranslatePosition** (int currTime, int axis)

Additional Inherited Members

4.19 com.nexstreaming.nexeditorsdk.nexAnimate.MoveTrackingPath Interface Reference

Public Member Functions

- float getTranslatePosition (int coordinate, float timeRatio)

4.19.1 Detailed Description

This interface is used as a parameter of move animation and rotate animation. As for move animation, this will be the parameter `path`, the location value for the method getMove(). As for rotate animation, this will be the parameter `center`, the pivot value for the method getRotate().

See Also

getMove(int, int, com.nexstreaming.nexeditorsdk.nexAnimate.MoveTrackingPath)
getRotate(int, int, boolean, int, com.nexstreaming.nexeditorsdk.nexAnimate.MoveTrackingPath)

Since

version 1.1.0

4.19.2 Member Function Documentation

4.19.2.1 float com.nexstreaming.nexeditorsdk.nexAnimate.MoveTrackingPath.getTranslatePosition (int *coordinate,* float *timeRatio*)

Parameters

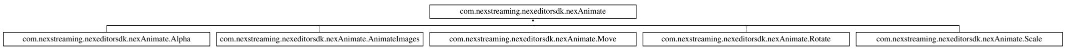
coordinate	• kCoordinateX = { kCoordinateX} : X-coordinate • kCoordinateY = { kCoordinateY} : Y-coordinate
timeRatio	In range of 0 to 1, where 0 is start time and 1 is end time of the animation effect.

Returns

The coordinate values returned, depending on the value of the parameter `coordinate`.

4.20 com.nexstreaming.nexeditorsdk.nexAnimate Class Reference

Inheritance diagram for com.nexstreaming.nexeditorsdk.nexAnimate:


com.nexstreaming.nexeditorsdk.nexAnimate
com.nexstreaming.nexeditorsdk.nexAnimate.Alpha
com.nexstreaming.nexeditorsdk.nexAnimate.AnimateImages
com.nexstreaming.nexeditorsdk.nexAnimate.Move
com.nexstreaming.nexeditorsdk.nexAnimate.Rotate
com.nexstreaming.nexeditorsdk.nexAnimate.Scale

Classes

- class Alpha
- class AnimateImages
- class Move
- interface MoveTrackingPath
- class Rotate
- class Scale

Public Member Functions

- void setTime (int startTime, int duration)
- nexAnimate setInterpolator (TimeInterpolator Interpolator)
- void resetFreeTypeAnimate ()

Static Public Member Functions

- static nexAnimate getAnimateImages (int startTime, int duration, int...ids)
- static nexAnimate getMove (int startTime, int duration, MoveTrackingPath path)
- static nexAnimate getAlpha (int startTime, int duration, float start, float end)
- static nexAnimate getRotate (int startTime, int duration, boolean clockwise, int rotateDegree, MoveTrackingPath center)
- static nexAnimate getScale (int startTime, int duration, float lastScaledX, float lastScaledY)
- static nexAnimate getScale (int startTime, int duration, float startScaledX, float startScaledY, float lastScaledX, float lastScaledY)
- static nexAnimate getScale (int startTime, int duration, MoveTrackingPath scale)

Static Public Attributes

- static final int kCoordinateX = 1
- static final int kCoordinateY = 2
- static final int kCoordinateZ = 3

Protected Member Functions

- **nexAnimate** (int startTime, int duration)
- float **timeRatio** (int time)
- int **getEndTime** ()
- float **getTranslatePosition** (int currTime, int axis)
- float **getAlpha** (int time)
- float **getAngleDegree** (int time, float startDegree, int axis)
- float **getScaledRatio** (int time, float startScaledRatio, int axis)
- boolean **onFreeTypeAnimate** (int time, nexOverlayItem overlay)

Protected Attributes

- int **mStartTime**
- int **mDuration**
- boolean **mRunning**
- int **mdX** = 0
- int **mdY** = 0
- int **mdZ** = 0
- float **mAlpha** = 1
- float **mScaledX** = 1
- float **mScaledY** = 1
- float **mScaledZ** = 1
- float **mRotateDegreeX** = 0
- float **mRotateDegreeY** = 0
- float **mRotateDegreeZ** = 0

4.20.1 Detailed Description

This class is used to set an animation effect of an overlay item.

This is a list of the animation effects available:

- Image animation : Prints sequence of images fast enough to make it look like the image is moving.
- Move animation : Moves an overlay item to a desired location on the screen.
- Rotate animation : Rotates an overlay item.
- Alpha animation : Sets the transparency of an overlay item.
- Scale animation : Sets the size of an overlay item.

The animation uses the linear functional equation and a time range of 0 to 1 to create an animation effect. The x-coordinate and y-coordinate indicate the center of the overlay item being animated.

See Also

com.nexstreaming.nexeditorsdk.nexOverlayItem

Since

version 1.1.0

nexEditorSDK API Limited Level	{ EditorGlobal::kOverlayAnimateLimited}

4.20.2 Member Function Documentation

4.20.2.1 static nexAnimate com.nexstreaming.nexeditorsdk.nexAnimate.getAlpha (int *startTime*, int *duration*, float *start*, float *end*) [static]

This method create an alpha animation with the parameter values of this method. This method starts with the alpha value set to parameter `start`, and end with the alpha value set to parameter `end`. Range of 0 to 1, where 0 is start time and 1 is end time of the animation effect, the alpha value changes gradually over time.

Parameters

startTime	The start time of the animation in `milliseconds`.
duration	The duration of the animation in `milliseconds.`
start	The alpha value of start time of the animation.
end	The alpha value of end time of the animation.

Returns

A newly created alpha animation.

Example code :

```
nexAnimate ap = nexAnimate.getAlpha(4000,2000,1,0);
    overlayItem.addAnimate(ap);
```

See Also

getRotate(int, int, boolean, int, com.nexstreaming.nexeditorsdk.nexAnimate.MoveTrackingPath)
getMove(int, int, com.nexstreaming.nexeditorsdk.nexAnimate.MoveTrackingPath)
getAnimateImages(int, int, int...)
getScale(int, int, float, float)

Since

version 1.1.0

4.20.2.2 static nexAnimate com.nexstreaming.nexeditorsdk.nexAnimate.getAnimateImages (int *startTime*, int *duration*, int... *ids*) [static]

This method creates an image animation with the parameter values of this method.

Parameters

startTime	The start time of the animation in `milliseconds`.
duration	The duration of the animation in `milliseconds`.
ids	The android resource ID of the image animation.

Returns

A newly created image animation.

Example code :

```
nexAnimate ani = nexAnimate.getAnimateImages(
        0,5000,R.drawable.myavatar_17004763_1
        ,R.drawable.myavatar_17004763_2
        ,R.drawable.myavatar_17004763_3
        ,R.drawable.myavatar_17004763_4
        ,R.drawable.myavatar_17004763_5
        ,R.drawable.myavatar_17004763_6
        ,R.drawable.myavatar_17004763_7
        ,R.drawable.myavatar_17004763_8
        ,R.drawable.myavatar_17004763_9
        ,R.drawable.myavatar_17004763_10
        );
 overlayItem.addAnimate(ani);
```

See Also

getRotate(int, int, boolean, int, com.nexstreaming.nexeditorsdk.nexAnimate.MoveTrackingPath)
getMove(int, int, com.nexstreaming.nexeditorsdk.nexAnimate.MoveTrackingPath)
getAlpha(int, int, float, float)
getScale(int, int, float, float)

Exceptions

NotSupportedAPILevel	nexEditorSDK API Limited Level is { EditorGlobal::kOverlayAnimateLimited}

Since

version 1.1.0

4.20.2.3 static nexAnimate com.nexstreaming.nexeditorsdk.nexAnimate.getMove (int *startTime,* int *duration,* MoveTrackingPath *path*) `[static]`

This method creates a move animation with the parameter values of this method. The overlay item will be moved to a location of coordinate values returned from the parameter `path`. Range of 0 to 1, where 0 is start time and 1 is end time of the animation effect, the overlay image moves gradually over time.

Parameters

startTime	The start time of the animation in `milliseconds`.
duration	The duration of the animation in `milliseconds`.
path	`MoveTrackingPath` to get the x-coordinate and y-coordinate value with given playtime of the animation.

Returns

A newly created move animation.

Example code :

```
nexAnimate mv = nexAnimate.getMove(0,5000,new nexAnimate.MoveTrackingPath(){
        public float getTranslatePosition(int coordinate, float timeRatio) {
            if( coordinate == nexAnimate.kCoordinateX ){
                return 700*timeRatio;
            } else if( coordinate == nexAnimate.kCoordinateY ) {
                return 400*timeRatio;
            }
            return 0;
        }
    });
    overlayItem.addAnimate(mv);
```

See Also

getRotate(int, int, boolean, int, com.nexstreaming.nexeditorsdk.nexAnimate.MoveTrackingPath)
getAlpha(int, int, float, float)
getAnimateImages(int, int, int...)
getScale(int, int, float, float)

Since

version 1.1.0

4.20.2.4 static nexAnimate com.nexstreaming.nexeditorsdk.nexAnimate.getRotate (int *startTime,* int *duration,* boolean *clockwise,* int *rotateDegree,* MoveTrackingPath *center*) [static]

This method creates a rotating animation.

The rotating animation created with this method will pivot on the coordinates that are set for the parameter `center`. Over the range of 0 to 1, where 0 is the start and 1 is the end of the animation effect, the overlay image rotates gradually over time.

Parameters

startTime	The start time of the animation in `msec` (milliseconds).
duration	The duration of the animation in `msec` (milliseconds).
clockwise	The rotation direction. Set to `TRUE` for clockwise or `FALSE` for counterclockwise.
rotateDegree	The number of rotation times, in degrees. (ex: To make one rotation = 360, and to make three rotations = 1080)
center	A `MoveTrackingPath` to provide the x- and y-coordinate values for the given playtime of the animation.

Returns

The newly created rotating animation.

Example code :

```
 nexAnimate ro = nexAnimate.getRotate(2000,3000,true,360,null);
overlayItem.addAnimate(ro);
```

See Also

getMove(int, int, com.nexstreaming.nexeditorsdk.nexAnimate.MoveTrackingPath)
getAlpha(int, int, float, float)
getAnimateImages(int, int, int...)
getScale(int, int, float, float)

Since

version 1.1.0

4.20.2.5 static nexAnimate com.nexstreaming.nexeditorsdk.nexAnimate.getScale (int *startTime,* int *duration,* float *lastScaledX,* float *lastScaledY*) [static]

This method creates a scaling animation.

A scaling animation changes the original size of an item by a scaling value set to both parameters `lastScaledX` and `lastScaledY`. Over the range of 0 to 1, where 0 is start time and 1 is end time of the animation effect, the overlay image changes its size gradually over time.

Parameters

startTime	The start time of the animation effect in `msec` (milliseconds).
duration	The duration of the animation effect in `msec` (milliseconds).
lastScaledX	The scaled width of the overlay item at the end of the animation.
lastScaledY	The scaled height of the overlay item at the end of the animation.

Returns

A newly created scaled animation.

Example code :

```
 nexAnimate sc = nexAnimate.getScale(0,3000,2,2);
overlayItem.addAnimate(sc);
```

See Also

getRotate(int, int, boolean, int, com.nexstreaming.nexeditorsdk.nexAnimate.MoveTrackingPath)
getMove(int, int, com.nexstreaming.nexeditorsdk.nexAnimate.MoveTrackingPath)
getAlpha(int, int, float, float)
getAnimateImages(int, int, int...)
getScale(int, int, float, float, float, float)
getScale(int, int, MoveTrackingPath)

Since

version 1.1.0

4.20.2.6 static nexAnimate com.nexstreaming.nexeditorsdk.nexAnimate.getScale (int *startTime,* int *duration,* float *startScaledX,* float *startScaledY,* float *lastScaledX,* float *lastScaledY*) [static]

This method ignores the set initial scaled values in `OverlayImage` and uses new initial scale values to create a scaled animation.

Example code :

```
nexAnimate sc = nexAnimate.getScale(0, expressDuration, 0, 0, 1, 1);
overlayItem.addAnimate(sc);
```

Parameters

startTime	The start time of the animation effect in `msec` (milliseconds).
duration	The duration of the animation effect in `msec` (milliseconds).
startScaledX	The scaled width of the overlay item at the start of the animation.
startScaledY	The scaled height of the overlay item at the start of the animation.
lastScaledX	The scaled width of the overlay item at the end of the animation.
lastScaledY	The scaled height of the overlay item at the end of the animation.

Returns

A newly created scaled animation.

See Also

getScale(int, int, float, float)
getScale(int, int, MoveTrackingPath)

Since

version 1.5.23

4.20.2.7 static nexAnimate com.nexstreaming.nexeditorsdk.nexAnimate.getScale (int *startTime,* int *duration,* MoveTrackingPath *scale*) [static]

This method is used to create a different scaled animation for each start time.

Example code :

```
 item.addAnimate(nexAnimate.getScale(0, duration, new nexAnimate.MoveTrackingPath() {
   int time = 0;
   @Override
   public float getTranslatePosition(int coordinate, float timeRatio) {
       if (coordinate == nexAnimate.kCoordinateX){
           return (float) (Math.sin((float)time/400f)*0.2f );
       }else if( coordinate == nexAnimate.kCoordinateY ){
           return (float) (Math.cos((float)time/400f)*0.2f );
       }else{
           time += 33;
       }
       return 1;
   }
})
```

Parameters

startTime	The start time of the animation effect in `msec` (milliseconds).
duration	The duration of the animation effect in `msec` (milliseconds).
scale	The scale values, which depends on the coodinate parameters that are passed in using the `MoveTrackingPath` interface.

Returns

A newly created scaled animation.

See Also

getScale(int, int, float, float, float, float)
getScale(int, int, float, float)

Since

version 1.5.23

4.20.2.8 void com.nexstreaming.nexeditorsdk.nexAnimate.resetFreeTypeAnimate ( )

This method initiates the variables that are related to animation movements.

Since

version 1.1.0

4.20.2.9 nexAnimate com.nexstreaming.nexeditorsdk.nexAnimate.setInterpolator (TimeInterpolator *Interpolator*)

In `nexAnimate`, the time interpolator increments by the ratio of 1:1 in the type of the first order linear equation.W

For a new time increment in a different type of interpolator other than linear, set a new interpolator using this method.

Example code :

```
 nexAnimate sc = nexAnimate.getAlpha(0, 1000, 0, 1).setInterpolator(new
      AccelerateDecelerateInterpolator());
overlayItem.addAnimate(sc);
```

Parameters

Interpolator	The Time Interpolator provided by the Android SDK API.

Returns

this. The set instance. Setting is available after calling the setInterpolator() that has been created as shown in the example above.

Since

version 1.5.23

4.20.2.10 void com.nexstreaming.nexeditorsdk.nexAnimate.setTime (int *startTime,* int *duration*)

This method sets the start time and the duration of an animation that's set to an overlay item.

Example code :

```
nexAnimate ani = nexAnimate.getAnimateImages(
        0, 5000, R.drawable.ilove0001
        , R.drawable.ilove0002
        , R.drawable.ilove0003
        , R.drawable.ilove0004
        , R.drawable.ilove0005
        , R.drawable.ilove0006
        , R.drawable.ilove0007
        , R.drawable.ilove0008
        , R.drawable.ilove0009
        , R.drawable.ilove0010
        );
ani.setTime(0, 2000);
```

Parameters

startTime	The start time of the animation in `milliseconds`.
duration	The duration of the animation in `milliseconds`.

Exceptions

com.nexstreaming.nexeditorsdk.exception.InvalidRangeException	

Since

version 1.1.0

4.20.3 Member Data Documentation

4.20.3.1 final int com.nexstreaming.nexeditorsdk.nexAnimate.kCoordinateX = 1 [static]

The value of the parameter `coordinate` of the interface `MoveTrackingPath`. This indicates the x-coordinate of the center of an overlay item.

See Also

com.nexstreaming.nexeditorsdk.nexAnimate.MoveTrackingPath

Since

version 1.1.0

4.20.3.2 final int com.nexstreaming.nexeditorsdk.nexAnimate.kCoordinateY = 2 [static]

The value of the parameter `coordinate` of the interface `MoveTrackingPath`. This indicates the y-coordinate.

See Also

com.nexstreaming.nexeditorsdk.nexAnimate.MoveTrackingPath

Since

version 1.1.0

4.20.3.3 final int com.nexstreaming.nexeditorsdk.nexAnimate.kCoordinateZ = 3 [static]

The value of the parameter `coordinate` of the interface `MoveTrackingPath`. This indicates the z-coordinate.

See Also

com.nexstreaming.nexeditorsdk.nexAnimate.MoveTrackingPath

Since

version 1.3.43

4.21 com.nexstreaming.nexeditorsdk.nexApplicationConfig Class Reference

Classes

- enum APILevel
- class OnEffectLoadListener

Static Public Member Functions

- static synchronized void createApp (Context Context)
- static void init (Context AppContext, String AppName)
- static void init (Context AppContext, String AppName, String SDK_PubKey)
- static void init (Context AppContext, String AppName, File PluginPath, String SDK_PubKey)
- static void init (Context AppContext, Context otherThemeContext, String AppName, File PluginPath, String SDK_PubKey)
- static void releaseApp ()
- static void waitForLoading (Context context, final Runnable listener)
- static void waitForMinimumLoading (Context context, final Runnable listener)
- static void setEffectLoadListener (OnEffectLoadListener listener)
- static String getSDKVersion ()
- static String getDevelopString ()
- static void setAspectMode (int aspectRatioMode)
- static float getAspectRatio ()
- static float getAspectRatioInScreenMode ()
- static int getAspectRatioMode ()
- static void setScreenMode (int mode)
- static int getScreenMode ()
- static final int getApiLevel ()
- static final boolean isSupportedApi (APILevel level)

Static Public Attributes

- static final int kAspectRatio_Mode_16v9 = 1
- static final int kAspectRatio_Mode_1v1 = 2
- static final int kAspectRatio_Mode_9v16 = 3
- static final int **kAspectRatio_Mode_2v1** = 4
- static final int kScreenMode_normal = 0
- static final int kScreenMode_horizonDual = 1

4.21.1 Detailed Description

This class initializes the NexEditor™ SDK. init(android.content.Context, String) } must be called first when creating the application. Since the NexEditor™ SDK operates using a Singleton class, createApp(android.content.Context) must be created when making the application.

Example code:

```
public class KMSDKApplication extends Application {
        public void onCreate() {
            nexApplicationConfig.createApp(this);
            nexApplicationConfig.init(getApplicationContext(),"application naming");
        }
    }
```

Since

version 1.0.0

4.21.2 Member Function Documentation

4.21.2.1 static synchronized void com.nexstreaming.nexeditorsdk.nexApplicationConfig.createApp (Context *Context*) [static]

This method creates a singleton instance. This method must be called only once before using the NexEditor™ SDK.

Example code:

```
public class KMSDKApplication extends Application {
     public KMSDKApplication() {
          super();
          nexApplicationConfig.createApp(this);
      }
 }
```

Parameters

Context	The Android context.

Since

version 1.0.0

4.21.2.2 static final int com.nexstreaming.nexeditorsdk.nexApplicationConfig.getApiLevel ( ) [static]

This method gets the supported API level.

Returns

The supported API level.

Since

version 1.5.19

4.21.2.3 static float com.nexstreaming.nexeditorsdk.nexApplicationConfig.getAspectRatio ( ) [static]

This method gets the aspect ratio of the output screen as a `float`.

Example code :

```
nexEffectPreviewView(Context context) {
        super(context);
        super.setAspectRatio(nexApplicationConfig.getAspectRatio());
    }
```

Returns

1 if set to `kAspectRatio_Mode_1v1`; 2 if set to `kAspectRatio_Mode_16v9`.

See Also

setAspectMode(int)
getAspectRatioMode()

Since

version 1.1.0

4.21.2.4 static float com.nexstreaming.nexeditorsdk.nexApplicationConfig.getAspectRatioInScreenMode ( ) [static]

This method gets the aspect ratio of the output screen. The screen width will be doubled in the dual screen mode.

Example code :

```
public nexEngineView(Context context) {
        super(context);
        super.setAspectRatio(nexApplicationConfig.getAspectRatioInScreenMode());
    }
```

Returns

The aspect ratio of the output screen in the format of width / height.

Since

version 1.3.52

4.21.2.5 static int com.nexstreaming.nexeditorsdk.nexApplicationConfig.getAspectRatioMode ( ) [static]

This method gets the aspect ratio of the output screen as an `integer`.

Example code :

```
if(nexApplicationConfig.getAspectRatioMode() ==
      nexApplicationConfig.kAspectRatio_Mode_1v1 )
```

Returns

1 if set to `kAspectRatio_Mode_1v1`; 2 if set to `kAspectRatio_Mode_16v9`.

Since

version 1.1.0

See Also

setAspectMode(int)
getAspectRatio()

4.21.2.6 static String com.nexstreaming.nexeditorsdk.nexApplicationConfig.getDevelopString ( ) [static]

This method gets the build version of the NexEditor™ SDK. The build version is a 4 digit number and it will be reset when the SDK version is updated.

Returns

`Official` if the version is official; otherwise `Develop`.

Since

version 1.3.0

4.21.2.7 static int com.nexstreaming.nexeditorsdk.nexApplicationConfig.getScreenMode ( ) [static]

This method gets the mode currently set to the output screen.

Example code :

```
mVideoEditor.setScreenMode(nexApplicationConfig.getScreenMode());
```

Returns

The screen mode currently set to output screen as an `integer`. 0 for normal screen mode or 1 for dual screen mode.

See Also

setScreenMode(int)

Since

version 1.3.52

4.21.2.8 static String com.nexstreaming.nexeditorsdk.nexApplicationConfig.getSDKVersion ( ) [static]

This method gets the version information of the NexEditor™ SDK as a `String`.

Returns

The version of the NexEditor™ SDK in use, in the format `Major.Minor.Patch`.

Since

version 1.0.0

See Also

getDevelopString()

4.21.2.9 static void com.nexstreaming.nexeditorsdk.nexApplicationConfig.init (Context *AppContext*, String *AppName*) [static]

This method initializes `EffectLibrary` and the cache directory. This method must be called once when starting the application.

Example code :

```
 public class KMSDKApplication extends Application {
   public void onCreate() {
       nexApplicationConfig.init(getApplicationContext(),"nexsdkapp");
       super.onCreate();
   }
}
```

Parameters

AppContext	The Android context.
AppName	External storage path name.

See Also

init(android.content.Context, android.content.Context, String, java.io.File, String)
init(android.content.Context, String, java.io.File, String)
init(android.content.Context, String, String)

Since

version 1.0.0

4.21.2.10 static void com.nexstreaming.nexeditorsdk.nexApplicationConfig.init (Context *AppContext,* String *AppName,* String *SDK_PubKey*) [static]

This method initializes `EffectLibrary` and the cache directory. This method must be called once when starting the application.

Parameters

AppContext	The Android context.
AppName	External storage path name.
SDK_PubKey	OEM Theme public key as a `String`.

See Also

init(android.content.Context, android.content.Context, String, java.io.File, String)
init(android.content.Context, String, java.io.File, String)
init(android.content.Context, String)

Since

version 1.3.0

4.21.2.11 static void com.nexstreaming.nexeditorsdk.nexApplicationConfig.init (Context *AppContext,* String *AppName,* File *PluginPath,* String *SDK_PubKey*) [static]

This method initializes `EffectLibrary` and the cache directory. This method must be called once when starting the application.

Parameters

AppContext	The Android context.
AppName	External storage path name.
PluginPath	The theme plug-in path. This path will be made automatically if set to `null`.

SDK_PubKey	OEM Theme public key as a `String`.

See Also

init(android.content.Context, android.content.Context, String, java.io.File, String)
init(android.content.Context, String, java.io.File, String)
init(android.content.Context, String)

Since

version 1.3.4

4.21.2.12 static void com.nexstreaming.nexeditorsdk.nexApplicationConfig.init (Context *AppContext,* Context *otherThemeContext,* String *AppName,* File *PluginPath,* String *SDK_PubKey*) [static]

This method initializes `EffectLibrary` and the cache directory. This method must be called once when starting the application.

Parameters

AppContext	The Android context.
otherThemeContext	Public android context from other APKs.
AppName	External storage path name.
PluginPath	The theme plug-in path. This path will be made automatically if set to `null`.
SDK_PubKey	OEM Theme public key as a `String`.

See Also

init(android.content.Context, String, String)
init(android.content.Context, String, java.io.File, String)
init(android.content.Context, String)

Since

version 1.3.4

4.21.2.13 static final boolean com.nexstreaming.nexeditorsdk.nexApplicationConfig.isSupportedApi (APILevel *level*) [static]

This method checks whether or not an API is available to use.

Example code :

```
if(isSupportedApi(nexApplicationConfig.APILevel.OverlayImageLimited)) {
 // ToDo.
} else {}
```

Parameters

level	API level to be compared.

Returns

`TRUE` if the API is available.

Since

version 1.5.19

See Also

com.nexstreaming.nexeditorsdk.nexApplicationConfig.APILevel::fromValue(int)

4.21.2.14 static void com.nexstreaming.nexeditorsdk.nexApplicationConfig.releaseApp ( ) [static]

This method clears the resources of NexEditor™ .

This method must be called when the activity is destroyed.

Since

1.1.0

4.21.2.15 static void com.nexstreaming.nexeditorsdk.nexApplicationConfig.setAspectMode (int *aspectRatioMode*) [static]

This method sets the aspect ratio of the output screen.

Example code :

```
nexApplicationConfig.setAspectMode(nexApplicationConfig.kAspectRatio_Mode_16v9
      );
```

Parameters

aspectRatioMode	The aspect ratio mode to set, as an `integer`. This will be either 1 for `kAspectRatio_Mode_16v9` or 2 for `kAspectRatio_Mode_1v1`.

See Also

getAspectRatioMode()
getAspectRatio()

Since

version 1.1.0

4.21.2.16 static void com.nexstreaming.nexeditorsdk.nexApplicationConfig.setEffectLoadListener (OnEffectLoadListener *listener*) [static]

If waitForLoading(Context, Runnable) or waitForMinimumLoading(Context, Runnable) is called, effect loading will begin. To check package completion of the effects being loaded, use this API.

Example code :

```
nexApplicationConfig.setEffectLoadListener(new nexApplicationConfig.OnEffectLoadListener() {
    public void onEffectPriorityLoadComplete() { }

    public void onEffectLoadComplete() { Log.d(LOG_TAG,"EffectLoadComplete"); }

    public void onEffectLoad(String s, int i) { Log.d(LOG_TAG,"onEffectLoad="+s); }
});
```

Parameters

A	listener OnEffectLoadListener

Since

1.5.42

4.21.2.17 static void com.nexstreaming.nexeditorsdk.nexApplicationConfig.setScreenMode (int *mode*) [static]

This method sets the mode for the output screen.

Example code :

```
nexApplicationConfig.setScreenMode(nexApplicationConfig.kScreenMode_normal
      );
```

Parameters

mode	The screen mode as an `integer`. 0 for normal screen mode or 1 for dual screen mode.

See Also

getScreenMode()

Since

version 1.3.52

4.21.2.18 static void com.nexstreaming.nexeditorsdk.nexApplicationConfig.waitForLoading (Context *context,* final Runnable *listener*) [static]

This method indicates the completion time of the effect library's initialization process with **Runnable**. The effect library can be used only after the initialization process has completed.

Example code :

```
nexApplicationConfig.waitForLoading(this, new Runnable() {
    public void run() {
        setContentView(R.layout.activity_main);
        nexEffectLibrary fxlib = nexEffectLibrary.getEffectLibrary(MainActivity.this);
        ......
    }
});
```

Parameters

context	The Android context.
listener	The runnable thread.

See Also

waitForMinimumLoading(android.content.Context, Runnable)

Since

version 1.0.0

4.21.2.19 static void com.nexstreaming.nexeditorsdk.nexApplicationConfig.waitForMinimumLoading (Context *context,* final Runnable *listener*) [static]

This method gets only the essential files needed for playback. when importing the entire theme and transition effect files takes too long.

Example code :

```
nexApplicationConfig.waitForMinimumLoading(this, new Runnable() {
      public void run() {
          setContentView(R.layout.activity_main);
          nexEffectLibrary fxlib = nexEffectLibrary.getEffectLibrary(MainActivity.this);
          ......
      }
  });
```

Parameters

context	The Android context.
listener	The runnable thread.

See Also

waitForLoading(android.content.Context, Runnable)

Since

version 1.3.43

4.21.3 Member Data Documentation

4.21.3.1 final int com.nexstreaming.nexeditorsdk.nexApplicationConfig.kAspectRatio_Mode_16v9 = 1 [static]

This indicates the possible mode of `kAspectRatio`. Set to this mode for a 16x9 screen ratio.

See Also

com.nexstreaming.nexeditorsdk.nexAnimate.MoveTrackingPath

Since

version 1.3.0

4.21.3.2 final int com.nexstreaming.nexeditorsdk.nexApplicationConfig.kAspectRatio_Mode_1v1 = 2 [static]

This indicates the possible mode of `kAspecRatio`. Set to this mode for a 1x1 screen ratio.

See Also

com.nexstreaming.nexeditorsdk.nexAnimate.MoveTrackingPath

Since

version 1.3.0

4.21.3.3 final int com.nexstreaming.nexeditorsdk.nexApplicationConfig.kAspectRatio_Mode_9v16 = 3 [static]

This indicates the possible mode of `kAspecRatio`. Set to this mode for a 9x16 screen ratio.

See Also

com.nexstreaming.nexeditorsdk.nexAnimate.MoveTrackingPath

Since

version 1.3.0

4.21.3.4 final int com.nexstreaming.nexeditorsdk.nexApplicationConfig.kScreenMode_horizonDual = 1 [static]

This indicates the possible mode of `kScreenMode`. Set to this mode for horizontal dual screen.

See Also

com.nexstreaming.nexeditorsdk.nexAnimate.MoveTrackingPath

Since

version 1.3.0

4.21.3.5 final int com.nexstreaming.nexeditorsdk.nexApplicationConfig.kScreenMode_normal = 0 [static]

This indicates the possible mode of `kScreenMode`. Set to this mode for normal screen.

See Also

com.nexstreaming.nexeditorsdk.nexAnimate.MoveTrackingPath

Since

version 1.3.0

4.22 com.nexstreaming.nexeditorsdk.nexAssetMediaManager Class Reference

Classes

- class AssetMedia

Public Member Functions

- void loadMedia (int clipType)
- String[] getAssetMediaIds (int clipType)
- nexAssetMediaManager.AssetMedia getAssetMedia (String Id)
- List
  < nexAssetMediaManager.AssetMedia > getAssetMedias ()
- nexAssetMediaManager.AssetMedia[] getAssetMedias (int type)
- nexClip createAudioClip (String Id)
- void applyProjectBGM (nexProject project, String Id)

Static Public Member Functions

- static nexAssetMediaManager getAudioManager (Context appContext)

4.22.1 Detailed Description

This is a class that manages effects or BGM media from Asset Items.

Since

1.7.1

4.22.2 Member Function Documentation

4.22.2.1 void com.nexstreaming.nexeditorsdk.nexAssetMediaManager.applyProjectBGM (nexProject *project,* String *Id*)

This sets the BGM of nexProject with the ID of a Media Asset item. If the ID is incorrect or it is not Audio, BGM is muted. Run loadMedia(int) at least once

Parameters

project	The project to which BGM will be set.
Id	Media Asset item ID, which must be Audio type.

Since

1.7.1

4.22.2.2 nexClip com.nexstreaming.nexeditorsdk.nexAssetMediaManager.createAudioClip (String *Id*)

This creates a nexClip with the ID of a Media Asset item. Run loadMedia(int) at least once

Parameters

Id	Media Asset item ID

Returns

`NULL` - loadMedia(int) did not run OR the ID is not one of the Media Asset Type IDs.

Since

1.7.1

4.22.2.3 nexAssetMediaManager.AssetMedia com.nexstreaming.nexeditorsdk.nexAssetMediaManager.getAssetMedia (String *Id*)

This gets AssetMedia with the ID of a Media Asset item. Run loadMedia(int) at least once

Parameters

Id	Asset Item ID

Returns

`NULL` - loadMedia(int) did not run OR the ID is not one of the Media Asset Type IDs.

Since

1.7.1

4.22.2.4 String [ ] com.nexstreaming.nexeditorsdk.nexAssetMediaManager.getAssetMediaIds (int *clipType*)

This gets the media item ID of a desired clip type. Run loadMedia(int) at least once

Parameters

clipType	nexClip.kCLIP_TYPE_AUDIO - audio, nexClip.kCLIP_TYPE_VIDEO - video, nexClip.kCLIP_TYPE_IMAGE - image

Returns

Array

Since

1.7.1

4.22.2.5 List<nexAssetMediaManager.AssetMedia> com.nexstreaming.nexeditorsdk.nexAssetMediaManager.getAssetMedias ( )

This gets a list of all Media Asset items. Run loadMedia(int) at least once

Returns

An uneditable List.

Since

1.7.1

4.22.2.6 nexAssetMediaManager.AssetMedia [] com.nexstreaming.nexeditorsdk.nexAssetMediaManager.getAssetMedias (int *type*)

This gets a list of Media Asset Items of a desire type as an `array`. Run loadMedia(int) at least once

Parameters

type	nexClip.kCLIP_TYPE_AUDIO - audio, nexClip.kCLIP_TYPE_VIDEO - video, nexClip.kCLIP_TYPE_IMAGE - image

Returns

Array

Since

1.7.1

4.22.2.7 static nexAssetMediaManager com.nexstreaming.nexeditorsdk.nexAssetMediaManager.getAudioManager (Context *appContext*) `[static]`

This gets an instance of nexAssetMediaManager.

Parameters

appContext	Input getApplicationContext(), but NOT `NULL`.

Returns

An instance of nexAssetMediaManager.

Since

1.7.1

4.22.2.8 void com.nexstreaming.nexeditorsdk.nexAssetMediaManager.loadMedia (int *clipType*)

This loads media from Asset DB to nexAssetMediaManager. This is only called once in the beginning; should also be called if the DB is updated.

Parameters

clipType	nexClip.kCLIP_TYPE_AUDIO - audio, nexClip.kCLIP_TYPE_VIDEO - video, nexClip.kCLIP_TYPE_IMAGE - image

Since

1.7.1

4.23 com.nexstreaming.nexeditorsdk.nexAssetPackageManager Class Reference

nexAssetPackageManager is a class that installs and manages any downloaded Assets from the Asset Store Application from NexStreaming.

Classes

- interface Asset
- class **AssetEnt**
- enum AssetInstallType
- enum Category
- interface Item
- class **ItemEnt**
- class OnInstallPackageListener
- enum PreAssetCategoryAlias
- interface RemoteAssetInfo
- class **RemoteAssetInfoEnt**

Public Member Functions

- List< Item > getInstalledAssetItems ()
- List< Item > getInstalledAssetItems (Category category)

- void getInstalledAssetItems (List< Item > input, Category category)
- List< Item > getInstalledAssetItemsByAssetIDx (int assetIdx)
- List< Item > getInstalledAssetItemsByAssetID (String assetId)
- List< Asset > getInstalledAsset ()
- List< Asset > getInstalledAssetByAppAsset ()
- List< Asset > getInstalledAssetByAppShare ()
- List< Asset > getInstalledAssetByStore ()
- Item getInstalledAssetItemById (String id)
- void uninstallPackageByAssetIdx (int assetIdx)
- void uninstallPackageById (String itemId)
- int findNewPackages ()
- boolean isInstallingPackages ()
- void installPackagesAsync (final OnInstallPackageListener listener)
- String[] getAssetCategoriesWithInstalledItems ()
- RemoteAssetInfo[] getRemoteAssetInfos (int mode)
- RemoteAssetInfo getRemoteAssetInfo (int mode)

Static Public Member Functions

- static nexAssetPackageManager getAssetPackageManager (Context appContext)
- static String getAssetPackageMediaPath (Context appContext, String ItemId)

Static Public Attributes

- static final int **Mode_Hot** = 1
- static final int **Mode_New** = 2

4.23.1 Detailed Description

nexAssetPackageManager is a class that installs and manages any downloaded Assets from the Asset Store Application from NexStreaming.

Assets downloaded from the Asset Store APP may not immediately work in nexEditorSDK. To use them properly, use nexAssetPackageManager to install them. Assets downloaded from the Asset Store APP are encrypted, so the downloads are not available for immediate use. Therefore, communication with the Asset Store APP is required.

For this step, run the Asset Store APP service using nexAssetStoreAppUtils.sendAssetStoreAppServiceIntent(Context). For your information, deleting the Asset Store APP when Assets are downloaded will make them unusable, and will not be usable even if the application is reinstalled. Reinstalling the Asset Store APP will delete all the Assets that were not installed.

The code below is for installation.

Example code : 1

```
barProgressDialog = new ProgressDialog(TemplateManagerActivity.this);
barProgressDialog.setTitle("Installing Asset Package ...");
barProgressDialog.setMessage("ready to install");
barProgressDialog.setProgressStyle(barProgressDialog.STYLE_HORIZONTAL);
barProgressDialog.setProgress(0);
barProgressDialog.setMax(100);
barProgressDialog.show();

nexAssetPackageManager.getAssetPackageManager(getApplicationContext()).
      installPackagesAsync(new nexAssetPackageManager.OnInstallPackageListener() {
```

```
    private int progress;
    private int count = 0;
    private int max = 0;
    private boolean updateMassage = false;

    public void onProgress(int countPackage, int totalPackages, int progressInstalling) {

        if( count != countPackage ){
           updateMassage = true;
        }
        count= countPackage;
        max = totalPackages;
        progress = progressInstalling;

        updateBarHandler.post(new Runnable() {

            public void run() {
               if( updateMassage ){
                   barProgressDialog.setMessage("install("+count+"/"+max+") in progress...");
               }
               barProgressDialog.setProgress(progress);
            }
        });
    }

   public void onCompleted(int event) {
       barProgressDialog.dismiss();
   }
});
```

Most Asset IDs share its concept with path. For example, read the examples below if you wish to use an audio Asset.

Example code : 2

```
List<nexAssetPackageManager.Item> audios = nexAssetPackageManager.getAssetPackageManager(
      getApplicationContext()).getInstalledAssetItems(nexAssetPackageManager.Category.audio);
if( audios.size() > 0 ){
   nexClip clip = new nexClip(audios.get(0).id());
}
```

If you wish to get the file path of audio and video Assets, use getAssetPackageMediaPath(Context, String). If you wish to get a Typeface with a font ID, use nexFont#getTypeface(Context, String). Input the font Asset ID in the second ID parameter. For instructions on how to use a Template, please see nexTemplateManager. For Stickers or Overlays, see nexOverlayPreset.

Since

1.7.0

4.23.2 Member Function Documentation

4.23.2.1 int com.nexstreaming.nexeditorsdk.nexAssetPackageManager.findNewPackages ( )

This checks if there is any new Template package.

Call this API first to install a package with installPackagesAsync().

Returns

The number of new packages to install.

Since

1.7.0

See Also

#installPackagesAsync(OnInstallPackageListener)

4.23.2.2 String [ ] com.nexstreaming.nexeditorsdk.nexAssetPackageManager.getAssetCategoriesWithInstalledItems ( )

This gets the category alias names of the currently installed Assets. For the category alias name of basic Assets, see PreAssetCategoryAlias.

Returns

category alias names of the currently installed Assets.

See Also

PreAssetCategoryAlias

Since

1.7.1

4.23.2.3 static nexAssetPackageManager com.nexstreaming.nexeditorsdk.nexAssetPackageManager.getAssetPackageManager (Context *appContext*) [static]

This works via nexAssetPackageManager SingleTon

In the beginning, it creates an instance, and after the second call, it gets the instance that has been made.

Parameters

appContext	Input getApplicationContext().

Returns

SingleTon instance of nexAssetPackageManager

Since

1.7.0

4.23.2.4 static String com.nexstreaming.nexeditorsdk.nexAssetPackageManager.getAssetPackageMediaPath (Context *appContext*, String *ItemId*) [static]

This gets the absolute path of the video or audio content included in the Asset. If the content is packed like an Asset or Zip, this API automatically copies the file to the cache space and returns that location. If the content file size is big, there might be blocking for a short time. Packages downloaded from the Asset Store are installed, so they notify the file location and therefore, there is no blocking. So putting Assets in the plugin folder is recommended in case where there are video or audio content in preloaded Assets.

Parameters

appContext	- Input getApplicationContext().
ItemId	- Item ID of the Audio or video included in the Asset.

Returns

The Absolute Path of content.

Since

1.7.0

4.23.2.5 List<**Asset**> com.nexstreaming.nexeditorsdk.nexAssetPackageManager.getInstalledAsset ( )

This gets all information of the installed Asset Package.

Returns

Since

1.7.0

4.23.2.6 List<**Asset**> com.nexstreaming.nexeditorsdk.nexAssetPackageManager.getInstalledAssetByAppAsset ( )

This gets all information of the Assets in the application from installed Asset Packages.

Returns

Since

1.7.0

4.23.2.7 List<**Asset**> com.nexstreaming.nexeditorsdk.nexAssetPackageManager.getInstalledAssetByAppShare ( )

This gets information of the dev shared folder from installed Asset Packages.

Returns

Since

1.7.0

4.23.2.8 List<Asset> com.nexstreaming.nexeditorsdk.nexAssetPackageManager.getInstalledAssetByStore ( )

From installed Asset Packages, this gets the installation information of the Asset downloaded from the store.

Returns

Since

1.7.0

4.23.2.9 Item com.nexstreaming.nexeditorsdk.nexAssetPackageManager.getInstalledAssetItemById (String *id*)

If the Asset ID is known, the developer can get item information with this API.

Parameters

id	

Returns

Item

Since

1.7.0

4.23.2.10 List<Item> com.nexstreaming.nexeditorsdk.nexAssetPackageManager.getInstalledAssetItems ( )

This gets the list of the installed Asset Items.

Returns

Since

1.7.0

4.23.2.11 List<Item> com.nexstreaming.nexeditorsdk.nexAssetPackageManager.getInstalledAssetItems (Category *category*)

This gets the list of the installed Asset Items.

Parameters

category	The category to search.

Returns

The item list that exists in the category.

Since

1.7.0

4.23.2.12 void com.nexstreaming.nexeditorsdk.nexAssetPackageManager.getInstalledAssetItems (List< Item > *input*, Category *category*)

This gets the list of the installed Asset Items.

Parameters

input	The list that will input existing items in a category.
category	Category to search.

Since

1.7.0

4.23.2.13 List<Item> com.nexstreaming.nexeditorsdk.nexAssetPackageManager.getInstalledAssetItemsByAssetID (String *assetId*)

This gets the list of items included in theAsset Package ID.

Parameters

assetId	

Returns

Since

1.7.0

4.23.2.14 List<Item> com.nexstreaming.nexeditorsdk.nexAssetPackageManager.getInstalledAssetItemsByAssetIDx (int *assetIdx*)

This gets the list of Items which have the same IDX of the Asset.

Some Asset includes just one Asset, but some like Template is a package of many Assets. In other words, AssetIdx is the package ID.

Parameters

assetIdx	AssetIdx Item#packageInfo()

Returns

The Item list packaged with AssetIdx.

Since

1.7.0

4.23.2.15 RemoteAssetInfo com.nexstreaming.nexeditorsdk.nexAssetPackageManager.getRemoteAssetInfo (int *mode*)

This gets a random piece of information from a featured Asset. Gets one piece at a time from Assets that are currently not installed. Returns `NULL` if it does not acquire any information.

Parameters

mode	Input the value of nexAssetPackageManager#Mode_Hot.

Returns

Since

1.7.0

4.23.2.16 RemoteAssetInfo [] com.nexstreaming.nexeditorsdk.nexAssetPackageManager.getRemoteAssetInfos (int *mode*)

This gets the information of a featured Asset in the Asset Store. To get information, the Asset Store App must be installed and service must be running. If the API does not acquire any information, the array size of the return value becomes 0.

Parameters

mode	Input the value of nexAssetPackageManager#Mode_Hot.

Returns

Since

1.7.0

4.23.2.17 void com.nexstreaming.nexeditorsdk.nexAssetPackageManager.installPackagesAsync (final OnInstallPackageListener *listener*)

This API installs Assets received from the Asset Store.

As a pre-requisite, run findNewPackages() first. If findNewPackages() finds a new package, then the developer should use it. installPackagesAsync is a single task, so check if it is running using isInstallingPackages() before using it.

Parameters

listener	OnInstallPackageListener

See Also

findNewPackages()
isInstallingPackages()

Since

1.7.0

4.23.2.18 boolean com.nexstreaming.nexeditorsdk.nexAssetPackageManager.isInstallingPackages ( )

installPackagesAsync is a single task. If it is running, it cannot be re-run until it is finished.

This API checks if installPackagesAsync is running or not.

Returns

`TRUE` - installPackagesAsync is running. `FALSE` - installPackagesAsync is ready.

See Also

#installPackagesAsync(OnInstallPackageListener)

Since

1.7.0

4.23.2.19 void com.nexstreaming.nexeditorsdk.nexAssetPackageManager.uninstallPackageByAssetIdx (int *assetIdx*)

This uninstalls the package that matches the Asset IDX.

Check with Item#isDelete() first whether or not the Asset can be uninstalled, then use the API.

Parameters

assetIdx	

Since

1.7.0

4.23.2.20 void com.nexstreaming.nexeditorsdk.nexAssetPackageManager.uninstallPackageById (String *itemId*)

This uninstalls the Asset package that matches the ID.

Check with Item#isDelete() first whether or not the Asset can be uninstalled, then use the API.

Warning

This does not only uninstall the item, but also other items that match the Asset IDX automatically.

Parameters

itemId	

Since

1.7.0

4.24 com.nexstreaming.nexeditorsdk.nexAssetStoreAppUtils Class Reference

Static Public Member Functions

- static void setVendor (String vendor)
- static void setMimeType (int mimeType)
- static void setMimeTypeExtra (String mimeType)
- static void sendAssetStoreAppServiceIntent (Context appContext)
- static boolean isServiceRunningCheck (Context appContext)
- static boolean isInstalledAssetStoreApp (Context appContext)
- static boolean isInstalledKineMaster (Context appContext)
- static int runAssetStoreApp (Activity activity, String assetIdx)
- static void moveGooglePlayAssetStoreLink (Context appContext)
- static void moveGooglePlayKineMaster (Context appContext)

Static Public Attributes

- static final int AssetStoreMimeType_Template = 0x1
- static final int AssetStoreMimeType_Effect = 0x2
- static final int AssetStoreMimeType_Transition = 0x4
- static final int AssetStoreMimeType_Audio = 0x8
- static final int AssetStoreMimeType_Filter = 0x10
- static final int AssetStoreMimeType_Background = 0x20

- static final int AssetStoreMimeType_Overlay = 0x40
- static final int AssetStoreMimeType_RenderItem = 0x80
- static final int AssetStoreMimeType_Font = 0x100
- static final int AssetStoreMimeType_TitleTemplate = 0x200
- static final int AssetStoreMimeType_Extra = 0x80000000

4.24.1 Detailed Description

In nexEditorSDK, new Asset such as Template, Effect , Sticker , Font , Audio , Filter can only be downloaded and installed from the Asset Store App.

nexAssetStoreAppUtils is composed of APIs that help calling the Asset Store App from NexStreaming while developing an app that uses nexEditorSDK.

Example code : 1 uses nexAssetStoreAppUtils and checks whether or not the Asset Store App is installed and currently running.

Example code : 1

```
if( nexAssetStoreAppUtils.isInstalledAssetStoreApp(getApplicationContext()) ){
          Log.d(TAG, "Asset store installed");
          if( !nexAssetStoreAppUtils.isServiceRunningCheck(getApplicationContext())) {
              Log.d(TAG, "Asset store service was not working");
              nexAssetStoreAppUtils.sendAssetStoreAppServiceIntent(getApplicationContext());
          }else{
              Log.d(TAG, "Asset store service was working");
          }
      }else{
          Log.d(TAG, "Asset store was not installed");
      }
```

Example code : 2 sends an Intent to the Asset Store App and runs it; this is the receiving part when the App is stopped. After getting a response, it uses nexAssetPackageManager to install the downloaded Assets from the Asset Store App.

Example code : 2

```
private int AssetStoreRequestCode = 0;
......
   Button bt_install = (Button) findViewById(R.id.button_template_man_install);
       bt_install.setOnClickListener(new View.OnClickListener() {
           public void onClick(View v) {
               AssetStoreRequestCode = nexAssetStoreAppUtils.runAssetStoreApp(TemplateManagerActivity.this
      ,null);
           }
       });
   .......

   protected void onActivityResult(int requestCode, int resultCode, Intent data) {
       super.onActivityResult(requestCode, resultCode, data);
       if(requestCode == AssetStoreRequestCode && resultCode == Activity.RESULT_OK) {
           Log.d(TAG,"onActivityResult from Asset Store");
           //use nexAssetPackageManager
       }
   }
```

4.24.2 Member Function Documentation

4.24.2.1 static boolean com.nexstreaming.nexeditorsdk.nexAssetStoreAppUtils.isInstalledAssetStoreApp (Context *appContext*) [static]

This checks whether or not the Asset Store App is installed on the device.

Parameters

appContext	The value of getApplicationContext()

Returns

true `TRUE` - App exists, `FALSE` - App does not exist.

Since

1.7.0

4.24.2.2 static boolean com.nexstreaming.nexeditorsdk.nexAssetStoreAppUtils.isInstalledKineMaster (Context *appContext*) `[static]`

This checks whether or not KineMaster is installed on the device.

Parameters

appContext	The value of getApplicationContext()

Returns

`TRUE` - App exists, `FALSE` - App does not exist.

Since

1.7.0

4.24.2.3 static boolean com.nexstreaming.nexeditorsdk.nexAssetStoreAppUtils.isServiceRunningCheck (Context *appContext*) `[static]`

This checks whether or not the Asset Store App is currently running.

Parameters

appContext	The value of getApplicationContext()

Returns

`TRUE` - Service running, `FALSE` - Service stopped.

Since

1.7.0

4.24.2.4 static void com.nexstreaming.nexeditorsdk.nexAssetStoreAppUtils.moveGooglePlayAssetStoreLink (Context *appContext*) `[static]`

This redirects to the VAssetStore link in GooglePlay.

Parameters

appContext	The value of getApplicationContext()

Since

1.7.0

4.24.2.5 static void com.nexstreaming.nexeditorsdk.nexAssetStoreAppUtils.moveGooglePlayKineMaster (Context *appContext*) [static]

This redirects to the KinemMaster link in GooglePlay.

Parameters

appContext	The value of getApplicationContext()

Since

1.7.0

4.24.2.6 static int com.nexstreaming.nexeditorsdk.nexAssetStoreAppUtils.runAssetStoreApp (Activity *activity,* String *assetIdx*) [static]

This runs AssetStoreApp.

Parameters

activity	The activity which will call this API.
assetIdx	If assetID does exists, this redirects to the Asset screen. If null is input, it redirects to a certain screen depending on the set values of setMimeType() and setVendor().

Returns

The requestCode value in response of void onActivityResult(int requestCode, int resultCode, Intent data) of the Activity that called this API after AssetStoreApp is stopped.

Since

1.7.0

4.24.2.7 static void com.nexstreaming.nexeditorsdk.nexAssetStoreAppUtils.sendAssetStoreAppServiceIntent (Context *appContext*) [static]

This runs the Asset Store App service.

Parameters

appContext	The value of getApplicationContext().

Since

1.7.0

4.24.2.8 static void com.nexstreaming.nexeditorsdk.nexAssetStoreAppUtils.setMimeType (int *mimeType*) [static]

This sets the mimeType which can be selected in the Asset Store App.

Parameters

mimeType	

Since

1.7.0

4.24.2.9 static void com.nexstreaming.nexeditorsdk.nexAssetStoreAppUtils.setMimeTypeExtra (String *mimeType*) [static]

The developer can choose a separately categorized Asset when running the Asset Store App.

To do this, setMimeType(AssetStoreMimeType_Extra) must be inserted; MimeType agreed with NexStreaming should be entered in the form of `String` using setMimeTypeExtra(String).

Parameters

mimeType	

See Also

setMimeType(int)
nexAssetStoreAppUtils::AssetStoreMimeType_Extra

Since

1.7.0

4.24.2.10 static void com.nexstreaming.nexeditorsdk.nexAssetStoreAppUtils.setVendor (String *vendor*) [static]

This sets the name of the vendor that uses nexEditorSDK. This value should be matched with Nexstreaming.

The Asset Store App screen may be different depending on this value.

Parameters

vendor	The name of the vendor that uses nexEditorSDK.

Since

1.7.0

4.24.3 Member Data Documentation

4.24.3.1 final int com.nexstreaming.nexeditorsdk.nexAssetStoreAppUtils.AssetStoreMimeType_Audio = 0x8 [static]

The developer can choose Audio when running the Asset Store App.

See Also

setMimeType(int)

Since

1.7.0

4.24.3.2 final int com.nexstreaming.nexeditorsdk.nexAssetStoreAppUtils.AssetStoreMimeType_Background = 0x20 [static]

The developer can choose a Background when running the Asset Store App.

See Also

setMimeType(int)

Since

1.7.0

4.24.3.3 final int com.nexstreaming.nexeditorsdk.nexAssetStoreAppUtils.AssetStoreMimeType_Effect = 0x2 [static]

The developer can choose a Clip Effect when running the Asset Store App.

See Also

setMimeType(int)

Since

1.7.0

4.24.3.4 final int com.nexstreaming.nexeditorsdk.nexAssetStoreAppUtils.AssetStoreMimeType_Extra = 0x80000000 [static]

The developer can choose a separately categorized Asset when running the Asset Store App.

To do this, setMimeType(AssetStoreMimeType_Extra) must be inserted; MimeType agreed with NexStreaming should be entered in the form of `String` using setMimeTypeExtra(String).

See Also

setMimeType(int)
setMimeTypeExtra(String)
runAssetStoreApp(Activity, String)

Since

1.7.0

4.24.3.5 final int com.nexstreaming.nexeditorsdk.nexAssetStoreAppUtils.AssetStoreMimeType_Filter = 0x10 [static]

The developer can choose a Filter when running the Asset Store App.

See Also

setMimeType(int)

Since

1.7.0

4.24.3.6 final int com.nexstreaming.nexeditorsdk.nexAssetStoreAppUtils.AssetStoreMimeType_Font = 0x100 [static]

The developer can choose a Font when running the Asset Store App.

See Also

setMimeType(int)

Since

1.7.1

4.24.3.7 final int com.nexstreaming.nexeditorsdk.nexAssetStoreAppUtils.AssetStoreMimeType_Overlay = 0x40 [static]

The developer can choose a Sticker when running the Asset Store App.

See Also

setMimeType(int)

Since

1.7.0

4.24.3.8 final int com.nexstreaming.nexeditorsdk.nexAssetStoreAppUtils.AssetStoreMimeType_RenderItem = 0x80 [static]

The developer can choose an enhanced effect when running the Asset Store App.

See Also

setMimeType(int)

Since

1.7.0

4.24.3.9 final int com.nexstreaming.nexeditorsdk.nexAssetStoreAppUtils.AssetStoreMimeType_Template = 0x1 [static]

The developer can choose a Template when running the Asset Store App.

See Also

setMimeType(int)

Since

1.7.0

4.24.3.10 final int com.nexstreaming.nexeditorsdk.nexAssetStoreAppUtils.AssetStoreMimeType_TitleTemplate = 0x200 [static]

The developer can choose a Title Effect when running the Asset Store App.

See Also

setMimeType(int)

Since

1.7.1

4.24.3.11 final int com.nexstreaming.nexeditorsdk.nexAssetStoreAppUtils.AssetStoreMimeType_Transition = 0x4 [static]

The developer can choose a Transition Effect when running the Asset Store App.

See Also

setMimeType(int)

Since

1.7.0

4.25 com.nexstreaming.nexeditorsdk.nexAudioEdit Class Reference

Public Member Functions

- void setPitch (int mPitch)
- int getPitch ()
- void setCompressor (int mCompressor)

  This method sets the compressor factor of the audio clip. The compressor factor which automatically maintains a constant volume level for all different kinds of sound sources such as movies, music, or mobile TV.
- int getCompressor ()
- void setProcessorStrength (int mProcessorStrength)
- int getProcessorStrength ()
- void setBassStrength (int mBassStrength)
- int getBassStrength ()
- void setMusicEffect (int mMusicEffect)
- int getMusicEffect ()
- int getPanLeft ()
- void setPanLeft (int panLeft)
- int getPanRight ()
- void setPanRight (int panRight)
- void setVoiceChangerFactor (int factor)
- int getVoiceChangerFactor ()

Static Public Attributes

- static final int kVoiceFactor_NONE = 0
- static final int kVoiceFactor_CHIPMUNK = 1
- static final int kVoiceFactor_ROBOT = 2
- static final int kVoiceFactor_DEEP = 3
- static final int kVoiceFactor_MODULATION = 4
- static final int kMusicEffect_NONE = 0
- static final int kMusicEffect_LIVE_CONCERT = 1
- static final int kMusicEffect_STEREO_CHORUS = 2
- static final int kMusicEffect_MUSIC_ENHANCER = 3

4.25.1 Detailed Description

Since

1.5.43 This class handles playback of voice change, pitch, compressor, music effect, and pan control options of an audio clip in NexEditor™ .

Example code 1:

```
mEngine.getProject().getClip(0,true).getnexAudioEdit().setPitch(2);
mEngine.play();
```

Example code 2:

```
if( project.getTotalClipCount(true) == 1 ){
    nexClip clip = project.getClip(0,true);
    if( clip.getClipType() == nexClip.kCLIP_TYPE_VIDEO ){
        if(clip.getTotalTime() > 12000 ) {
            clip.getVideoClipEdit().setAutoTrim(nexVideoClipEdit.kAutoTrim_Divided, 3);
        }
    }
}
```

Since

version 1.5.43

4.25.2 Member Function Documentation

4.25.2.1 int com.nexstreaming.nexeditorsdk.nexAudioEdit.getBassStrength ( )

This method gets the bass strength, the second parameter, when applying a music effect to an audio clip.

Returns

See Also

setBassStrength(int)

Since

1.5.43

4.25.2.2 int com.nexstreaming.nexeditorsdk.nexAudioEdit.getCompressor ( )

This method gets the compressor factor of the audio clip.

Returns

See Also

setCompressor(int)

Since

1.5.43

4.25.2.3 int com.nexstreaming.nexeditorsdk.nexAudioEdit.getMusicEffect ( )

This method gets the music effect values applied to the audio clip.

Parameters

factor	Audio music effect as an `integer`, in the range of 0 ∼ 4. Possible values for music effect factor : • kMusicEffect_NONE : Default • kMusicEffect_LIVE_CONCERT • kMusicEffect_STEREO_CHORUS • kMusicEffect_MUSIC_ENHANCER

Since

version 1.5.43

See Also

setMusicEffect(int)

4.25.2.4 int com.nexstreaming.nexeditorsdk.nexAudioEdit.getPanLeft ( )

This method applies the pan left value to an audio clip.

Returns

The pan left factor, as an `integer` (-100 ∼ 100).

See Also

setPanLeft(int)

Since

1.5.43

4.25.2.5 int com.nexstreaming.nexeditorsdk.nexAudioEdit.getPanRight ( )

This method applies the pan right value to an audio clip.

Returns

The pan right factor, as an `integer` (-100 ∼ 100).

Since

version 1.5.43

See Also

setPanRight(int)

Since

1.5.43

4.25.2.6 int com.nexstreaming.nexeditorsdk.nexAudioEdit.getPitch ( )

This method gets the applied pitch factor of the audio clip.

Returns

See Also

setPitch(int)

Since

1.5.43

4.25.2.7 int com.nexstreaming.nexeditorsdk.nexAudioEdit.getProcessorStrength ( )

This method gets the processor strength, the first parameter, when applying a music effect to an audio clip.

Returns

See Also

setProcessorStrength(int)

Since

1.5.43

4.25.2.8 int com.nexstreaming.nexeditorsdk.nexAudioEdit.getVoiceChangerFactor ( )

This method gets the voice changer factor set to a clip.

Returns

The voice changer factor, as an `integer` (0 ∼ 4).

- kVoiceFactor_NONE : Default
- kVoiceFactor_CHIPMUNK : Chipmunk
- kVoiceFactor_ROBOT : Robot
- kVoiceFactor_DEEP : Deep
- kVoiceFactor_MODULATION : Modulation

Since

version 1.5.15

See Also

setVoiceChangerFactor(int)

4.25.2.9 void com.nexstreaming.nexeditorsdk.nexAudioEdit.setBassStrength (int *mBassStrength*)

This method sets the bass strength, the second parameter, when applying a music effect to an audio clip.

Parameters

mBassStrength	: 0 ∼ 6

See Also

getBassStrength()

Since

1.5.43

4.25.2.10 void com.nexstreaming.nexeditorsdk.nexAudioEdit.setCompressor (int *mCompressor*)

This method sets the compressor factor of the audio clip. The compressor factor which automatically maintains a constant volume level for all different kinds of sound sources such as movies, music, or mobile TV.

Parameters

mCompressor	: 0 ∼ 7

See Also

getCompressor()

Since

1.5.43

4.25.2.11 void com.nexstreaming.nexeditorsdk.nexAudioEdit.setMusicEffect (int *mMusicEffect*)

This method applies the music effect values to an audio clip.

Parameters

factor	Audio music effect as an `integer`, in the range of 0 ∼ 4. Possible values for music effect factor : • kMusicEffect_NONE : Default • kMusicEffect_LIVE_CONCERT • kMusicEffect_STEREO_CHORUS • kMusicEffect_MUSIC_ENHANCER

Since

version 1.5.43

See Also

getMusicEffect()

4.25.2.12 void com.nexstreaming.nexeditorsdk.nexAudioEdit.setPanLeft (int *panLeft*)

This method gets the pan left value of an audio clip.

Parameters

panLeft	: The left channel's pan value.

See Also

getPanLeft()

Since

1.5.43

4.25.2.13 void com.nexstreaming.nexeditorsdk.nexAudioEdit.setPanRight (int *panRight*)

This method gets the pan right value of an audio clip.

Parameters

panRight	: The right channel's pan value.

See Also

getPanRight()

Since

1.5.43

4.25.2.14 void com.nexstreaming.nexeditorsdk.nexAudioEdit.setPitch (int *mPitch*)

This method sets the pitch factor of the audio clip.

Parameters

mPitch	: -12 ∼ 12

See Also

getPitch()

Since

1.5.43

4.25.2.15 void com.nexstreaming.nexeditorsdk.nexAudioEdit.setProcessorStrength (int *mProcessorStrength*)

This method sets the processor strength, the first parameter, when applying a music effect to an audio clip.

Parameters

mProcessorStrength	: 0 ∼ 6

See Also

getProcessorStrength()

Since

1.5.43

4.25.2.16 void com.nexstreaming.nexeditorsdk.nexAudioEdit.setVoiceChangerFactor (int *factor*)

This method sets the voice changer factor which is used to modulate the audio clip.

Parameters

factor	Audio modulate value as an `integer`, in the range of 0 ∼ 4. Possible values for the voice changer factor : • kVoiceFactor_NONE : Default • kVoiceFactor_CHIPMUNK : Chipmunk • kVoiceFactor_ROBOT : Robot • kVoiceFactor_DEEP : Deep • kVoiceFactor_MODULATION : Modulation

Since

version 1.5.15

See Also

getVoiceChangerFactor()

4.25.3 Member Data Documentation

4.25.3.1 final int com.nexstreaming.nexeditorsdk.nexAudioEdit.kMusicEffect_LIVE_CONCERT = 1 `[static]`

See Also

setMusicEffect(int)
getMusicEffect()

Since

1.5.43

4.25.3.2 final int com.nexstreaming.nexeditorsdk.nexAudioEdit.kMusicEffect_MUSIC_ENHANCER = 3 `[static]`

See Also

setMusicEffect(int)
getMusicEffect()

Since

1.5.43

4.25.3.3 final int com.nexstreaming.nexeditorsdk.nexAudioEdit.kMusicEffect_NONE = 0 `[static]`

See Also

setMusicEffect(int)
getMusicEffect()

Since

1.5.43

4.25.3.4 final int com.nexstreaming.nexeditorsdk.nexAudioEdit.kMusicEffect_STEREO_CHORUS = 2 [static]

See Also

setMusicEffect(int)
getMusicEffect()

Since

1.5.43

4.25.3.5 final int com.nexstreaming.nexeditorsdk.nexAudioEdit.kVoiceFactor_CHIPMUNK = 1 [static]

See Also

setVoiceChangerFactor(int)
getVoiceChangerFactor()

Since

1.5.43

4.25.3.6 final int com.nexstreaming.nexeditorsdk.nexAudioEdit.kVoiceFactor_DEEP = 3 [static]

See Also

setVoiceChangerFactor(int)
getVoiceChangerFactor()

Since

1.5.43

4.25.3.7 final int com.nexstreaming.nexeditorsdk.nexAudioEdit.kVoiceFactor_MODULATION = 4 [static]

See Also

setVoiceChangerFactor(int)
getVoiceChangerFactor()

Since

1.5.43

4.25.3.8 final int com.nexstreaming.nexeditorsdk.nexAudioEdit.kVoiceFactor_NONE = 0 [static]

See Also

setVoiceChangerFactor(int)
getVoiceChangerFactor()

Since

1.5.43

4.25.3.9 final int com.nexstreaming.nexeditorsdk.nexAudioEdit.kVoiceFactor_ROBOT = 2 [static]

See Also

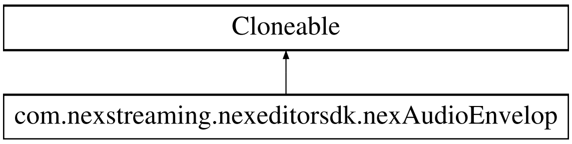
setVoiceChangerFactor(int)
getVoiceChangerFactor()

Since

1.5.43

4.26 com.nexstreaming.nexeditorsdk.nexAudioEnvelop Class Reference

Inheritance diagram for com.nexstreaming.nexeditorsdk.nexAudioEnvelop:



Public Member Functions

- void updateTrimTime (int start, int end)
- int addVolumeEnvelope (int time, int level)
- void changeVolumeLevelValue (int index, int level)
- void removeVolumeEnvelop ()
- void removeVolumeEnvelop (int index)
- int getVolumeEnvelopeTime (int index)
- int getVolumeEnvelopeLevel (int index)
- int getVolumeEnvelopeTimeAdj (int index)
- int getVolumeEnvelopeLength ()
- int[] getVolumeEnvelopeTimeList ()
- int[] getVolumeEnvelopeLevelList ()

Static Protected Member Functions

- static nexAudioEnvelop **clone** (nexAudioEnvelop src)

4.26.1 Detailed Description

This class allows to control the audio volume for each divided sections of audio item.

Example code:

```
setEnv.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        mTime.add(envTime.getText().toString());
        mVol.add(envVol.getText().toString());
        mItemAdp.notifyDataSetChanged();
    }
});
for(int i=0; i<mTime.size(); i++){
    project.getClip(0,false).getAudioEnvelop().addVolumeEnvelope(Integer.parseInt(mTime.get(i)),
      Integer.parseInt(mVol.get(i)));
}
```

4.26.2 Member Function Documentation

4.26.2.1 int com.nexstreaming.nexeditorsdk.nexAudioEnvelop.addVolumeEnvelope (int *time*, int *level*)

This method adds a volume envelope by setting the time and the volume values. The added volume envelopes are automatically listed by time.

Example code:

```
int index = project.getClip(0,false).getAudioEnvelop().addVolumeEnvelope(Integer.parseInt(mTime.get(i)),
      Integer.parseInt(mVol.get(i)));
```

Parameters

time	The time to add the volume envelop.
level	The audio volume to set to the envelop.

Returns

Index of the volume envelop list.

See Also

removeVolumeEnvelop(int)

Since

version 1.3.52

4.26.2.2 void com.nexstreaming.nexeditorsdk.nexAudioEnvelop.changeVolumeLevelValue (int *index,* int *level*)

This method changes the volume of an envelope on the location set with the parameter `index`.

Example code:

```
mClip.getAudioEnvelop().changeVolumeLevelValue(0, 200);
```

Parameters

index	The location of a volume envelop.
level	The new volume value to set to the envelop.

See Also

addVolumeEnvelope(int, int)
removeVolumeEnvelop(int)

Since

version 1.1.0

4.26.2.3 int com.nexstreaming.nexeditorsdk.nexAudioEnvelop.getVolumeEnvelopeLength ( )

This method gets the total number of envelopes set to an audio item.

Example code:

```
int totalcount = mClip.getAudioEnvelop().getVolumeEnvelopeLength();
```

Returns

The number of envelops.

Since

version 1.1.0

4.26.2.4 int com.nexstreaming.nexeditorsdk.nexAudioEnvelop.getVolumeEnvelopeLevel (int *index*)

This method gets the volume of an envelope that's on the location set with the parameter `index`.

Example code:

```
int level = mClip.getAudioEnvelop().getVolumeEnvelopelevel(0);
```

Parameters

index	The location of a volume envelop.

Returns

The volume level of the envelop or -1 if there is no envelop list.

Since

version 1.1.0

See Also

getVolumeEnvelopeTime(int)

4.26.2.5 int [ ] com.nexstreaming.nexeditorsdk.nexAudioEnvelop.getVolumeEnvelopeLevelList ( )

This method gets the volume level list of the envelopes set to an audio item.

Example code:

```
int volumelists = mClip.getAudioEnvelop().getVolumeEnvelopeLevelList();
```

Returns

The volume level list of the envelops, VolumeEnvelopeLevelList(int[ ])

See Also

getVolumeEnvelopeTimeList()

Since

version 1.3.52

4.26.2.6 int com.nexstreaming.nexeditorsdk.nexAudioEnvelop.getVolumeEnvelopeTime (int *index*)

This method gets the timestamp of an envelope that's on the location set with the parameter `index`.

Example code:

```
int time = mClip.getAudioEnvelop().getVolumeEnvelopeTime(0);
```

Parameters

index	The location of a volume envelop.

Returns

The timestamp of the envelop or -1 if there is no envelop list.

See Also

getVolumeEnvelopeLevel(int)

Since

version 1.1.0

4.26.2.7 int com.nexstreaming.nexeditorsdk.nexAudioEnvelop.getVolumeEnvelopeTimeAdj (int *index*)

This method gets the duration between the start time of a trimmed section and the time of an envelope's location set with the parameter `index`.

Example code:

```
int adjtime = mClip.getAudioEnvelop().getVolumeEnvelopeTimeAdj(0);
```

Parameters

index	The location of a volume envelop.

Returns

The duration between the start trim time and the envelope's location, otherwise -1 if there is no envelop list.

Since

version 1.1.0

See Also

getVolumeEnvelopeTime(int)

4.26.2.8 int [ ] com.nexstreaming.nexeditorsdk.nexAudioEnvelop.getVolumeEnvelopeTimeList ( )

This method gets the time list of the envelopes set to an audio item.

Example code:

```
int[] timelist = mClip.getAudioEnvelop().getVolumeEnvelopeTimeList();
```

Returns

The time list of envelops, VolumeEnvelopeTimeList(int[ ]).

See Also

getVolumeEnvelopeLevelList()

Since

version 1.3.52

4.26.2.9 void com.nexstreaming.nexeditorsdk.nexAudioEnvelop.removeVolumeEnvelop ( )

This method removes all the volume envelopes set on a clip.

Example code:

```
mClip.getAudioEnvelop().removeVolumeEnvelop();
```

See Also

addVolumeEnvelope(int, int)
changeVolumeLevelValue(int, int)
removeVolumeEnvelop(int)

Since

version 1.1.0

4.26.2.10 void com.nexstreaming.nexeditorsdk.nexAudioEnvelop.removeVolumeEnvelop (int *index*)

This method removes a volume envelope that's passed through `index` from the clip.

Example code:

```
mClip.getAudioEnvelop().removeVolumeEnvelop(0);
```

Parameters

index	The location of a volume envelop.

See Also

addVolumeEnvelope(int, int)
changeVolumeLevelValue(int, int)

Since

version 1.1.0

4.26.2.11 void com.nexstreaming.nexeditorsdk.nexAudioEnvelop.updateTrimTime (int *start,* int *end*)

This method updates the trim time of an audio item.

Example code:

```
mClip.getAudioEnvelop().updateTrimTime(0,mClip.getTotalTime());
```

Parameters

start	The start time of the trimmed section on the audio item.
end	The end time of the trimmed section on the audio item.

Since

version 1.3.52

4.27 com.nexstreaming.nexeditorsdk.nexAudioItem Class Reference

Public Member Functions

- nexClip getClip ()
- int getId ()
- int getStartTime ()
- int getEndTime ()
- void setTrim (int startTime, int endTime)
- void removeTrim ()
- int getStartTrimTime ()
- int getEndTrimTime ()

Protected Member Functions

- **nexAudioItem** (nexClip clip, int startTime, int endTime)
- void **setProjectTime** (int startTime, int endTime)

Protected Attributes

- nexClip **mClip**
- int **mTrimStartDuration**
- int **mTrimEndDuration**

4.27.1 Detailed Description

This class defines and manages audio items that are applied to a project with the NexEditor™ SDK.

The volume, start time, and end time of an audio item can be set and adjusted, and an audio item can be turned on and off.

For Example, an audio item can be added to a NexEditor™ project with the following sample code:

```
mSecondaryItems.add(new nexAudioItem(clip,startTime,endTime));
```

Example code :

```
bt.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            project.changeAudio(project.getAudioItem(0), mStartTime, project.getTotalTime());
            project.getAudioItem(0).setTrim(mTrimTime, project.getTotalTime());
            project.getAudioItem(0).getClip().setClipVolume(200);
        }
    }
```

Since

version 1.1.0

4.27.2 Member Function Documentation

4.27.2.1 nexClip com.nexstreaming.nexeditorsdk.nexAudioItem.getClip ( )

This method gets an audio item as an audio clip into the project.

Example code :

```
project.getAudioItem(0).getClip().setClipVolume(200);
```

Returns

An audio clip.

Since

version 1.3.37

See Also

getId()

4.27.2.2 int com.nexstreaming.nexeditorsdk.nexAudioItem.getEndTime ( )

This method gets the end time of an audio item in a project.

Example code :

```
for(int i = 0; i<mProject.getTotalClipCount(false); i++){
     if(duration < mProject.getAudioItem(i).getEndTime()){
         duration += mProject.getAudioItem(i).getEndTime();
     }
 }
```

Returns

The end time of the audio item, in `msec` (milliseconds).

See Also

getStartTime()

Since

version 1.1.0

4.27.2.3 int com.nexstreaming.nexeditorsdk.nexAudioItem.getEndTrimTime ( )

This method gets the new end time of an audio item in a project after it has been trimmed.

Example code :

```
int endtrimTime = project.getAudioItem(0).getEndTrimTime();
```

Returns

The end time of the trimmed audio item, in `msec` (milliseconds).

See Also

setTrim(int, int)
getStartTrimTime()

Since

version 1.1.0

4.27.2.4 int com.nexstreaming.nexeditorsdk.nexAudioItem.getId ( )

This method gets the ID of an audio item.

Example code :

```
for(int j = 0; j<m_listfilepath.size(); j++) {
    if(m_listfilepath.get(j).getID() == id) {
        m_listfilepath.get(j).setTimeline(2);
    }
}
```

Returns

The ID of an audio item.

See Also

getClip()

Since

version 1.5.0

4.27.2.5 int com.nexstreaming.nexeditorsdk.nexAudioItem.getStartTime ( )

This method gets the start time of an audio item in a project.

Example code :

```
TextView textView2 = new TextView(AudioTrackActivity.this);
   weight = project.getAudioItem(i).getStartTime()/1000;
   textView2.setLayoutParams(new LinearLayout.LayoutParams(ActionBar.LayoutParams.WRAP_CONTENT,
      ActionBar.LayoutParams.WRAP_CONTENT, weight));
```

Returns

The start time of the audio item, in `msec` (milliseconds).

See Also

getEndTime()

Since

version 1.1.0

4.27.2.6 int com.nexstreaming.nexeditorsdk.nexAudioItem.getStartTrimTime ( )

This method gets the new start time of an audio item in a project after it has been trimmed.

Example code :

```
int starttrimTime = project.getAudioItem(0).getStartTrimTime();
```

Returns

The start time of the trimmed audio item, in `msec` (milliseconds).

See Also

setTrim(int, int)
getEndTrimTime()

Since

version 1.1.0

4.27.2.7 void com.nexstreaming.nexeditorsdk.nexAudioItem.removeTrim ( )

This method removes the trim time set to an audio item.

Example code :

```
project.getAudioItem(0).removeTrim();
```

See Also

setTrim(int, int)

Since

version 1.3.52

4.27.2.8 void com.nexstreaming.nexeditorsdk.nexAudioItem.setTrim (int *startTime,* int *endTime*)

This method trims an existing audio item in a project, setting new start and end times for the item.

Example code :

```
project.getAudioItem(0).setTrim(mTrimTime, project.getTotalTime());
```

Parameters

startTime	The start time of the trimmed audio item, in `msec` (milliseconds).
endTime	The end time of the trimmed audio item, in `msec` (milliseconds).

See Also

getStartTrimTime()
getEndTrimTime()

Since

version 1.1.0

4.28 com.nexstreaming.nexeditorsdk.nexChecker Class Reference

Classes

- interface nexCheckerListener

Static Public Member Functions

- static void checkUHD (final nexCheckerListener checkerListener)

4.28.1 Detailed Description

This class is used to test a device's hardware encoder performance in runtime. The hardware codec resources is used for the encoder test. The test result will not return instantly, therefore the encoder test result should be received asynchronously.

Since

version 1.3.43

4.28.2 Member Function Documentation

4.28.2.1 static void com.nexstreaming.nexeditorsdk.nexChecker.checkUHD (final nexCheckerListener *checkerListener*) [static]

This method checks whether a device can encode UHD content.

Example code:

class sample code

```
nexChecker.checkUHD(new nexChecker.nexCheckerListener() {
            @Override
            public void onCheckerCapsResult(int result) {

            }
        });
```

Parameters

checkerListener	The listener to receive the result.

Since

version 1.3.43

This method gets the information when checking the current availability of the hardware encoder.

Parameters

checker	The checker class.
msg	The message about the encoder availability as a `string`.

Since

version 1.3.43

This method gets the result when checking the current availability of the hardware encoder.

Parameters

checker	The checker class.
nResult	0 if the encoder is available, otherwise 1.

Since

version 1.3.43

4.29 com.nexstreaming.nexeditorsdk.nexChecker.nexCheckerListener Interface Reference

Public Member Functions

- void onCheckerCapsResult (int result)

4.29.1 Detailed Description

This listener receives the device's hardware encoder test result from the `nexChecker`.

See Also

#checkUHD(nexCheckerListener)

Since

version 1.3.43

4.29.2 Member Function Documentation

4.29.2.1 void com.nexstreaming.nexeditorsdk.nexChecker.nexCheckerListener.onCheckerCapsResult (int *result*)

This method checks whether the hardware encoder of a device is currently available.

Parameters

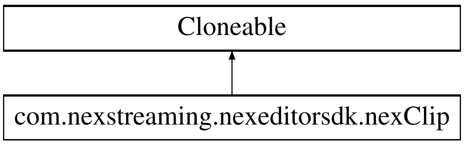
result	0 if currently available, otherwise 1.

Since

version 1.3.43

4.30 com.nexstreaming.nexeditorsdk.nexClip Class Reference

Inheritance diagram for com.nexstreaming.nexeditorsdk.nexClip:



Classes

- class ClipInfo
- class OnGetVideoClipDetailThumbnailsListener
- class OnGetVideoClipIDR2YOnlyThumbnailsListener
- class OnLoadVideoClipThumbnailListener

Public Member Functions

- boolean setBrightness (int value)
- boolean setContrast (int value)
- boolean setSaturation (int value)
- void setColorEffect (nexColorEffect effect)
- nexColorEffect getColorEffect ()
- int getBrightness ()
- int getContrast ()
- int getSaturation ()
- nexClip (String path)
- nexClip (String path, boolean useCache)
- int replaceClip (String path)
- int getFramesPerSecond ()
- boolean hasAudio ()
- boolean hasVideo ()
- String getPath ()
- int getClipType ()
- int getTotalTime ()
- int getWidth ()
- int getHeight ()
- int **getDisplayWidth** ()
- int **getDisplayHeight** ()
- int getSupportedResult ()
- int getAVCProfile ()
- int getAVCLevel ()
- int getRotateInMeta ()
- int getVideoBitrate ()
- int getAudioBitrate ()
- int getVideoDuration ()
- int getAudioDuration ()
- int getSeekPointCount ()
- int getSeekPointInterval ()
- int **getVideoRenderMode** ()
- nexClipEffect getClipEffect ()
- nexTransitionEffect getTransitionEffect ()
- nexVideoClipEdit getVideoClipEdit ()
- void setImageClipDuration (int ms)
- int getImageClipDuration ()
- int getProjectStartTime ()
- int getProjectEndTime ()
- void setAudioOnOff (boolean on)

- boolean getAudioOnOff ()
- void setMainThumbnail (Bitmap thumb)
- Bitmap getMainThumbnail (float height, float scaleDIPtoPX)
- int[] getVideoClipTimeLineOfThumbnail ()
- Bitmap getVideoClipTimeLineThumbnail (int rotation, int time, boolean fliph, boolean flipv)
- int loadVideoClipThumbnails (final OnLoadVideoClipThumbnailListener listener)
- int getVideoClipIFrameThumbnails (int thumbWidth, int thumbHeight, final OnGetVideoClipDetailThumbnailsListener listener)
- int getVideoClipIDR2YOnlyThumbnails (int thumbWidth, int thumbHeight, final int startTime, final int endTime, int maxCount, final OnGetVideoClipIDR2YOnlyThumbnailsListener listener)
- int getVideoClipDetailThumbnails (int thumbWidth, int thumbHeight, final int startTime, final int endTime, int maxCount, int rotate, final OnGetVideoClipDetailThumbnailsListener listener)
- int getVideoClipDetailThumbnails (int thumbWidth, int thumbHeight, int[] arrayTimeTab, int rotate, final OnGetVideoClipDetailThumbnailsListener listener)
- int getVideoClipDetailThumbnails (int thumbWidth, int thumbHeight, final int startTime, final int endTime, int maxCount, int rotate, boolean noCache, int[] arrayTimeTab, final OnGetVideoClipDetailThumbnailsListener listener)
- void setRotateDegree (int degree)
- int getRotateDegree ()
- nexCrop getCrop ()
- boolean setBGMVolume (int volume)
- int getBGMVolume ()
- boolean setClipVolume (int volume)
- int getClipVolume ()
- int[] getSeekPointsSync ()
- boolean isSolid ()
- boolean setSolidColor (int color)
- int getSolidColor ()
- nexAudioEnvelop getAudioEnvelop ()
- nexClipEffect getClipEffect (boolean internal)
- nexTransitionEffect getTransitionEffect (boolean internal)
- void setVoiceChangerFactor (int factor)
- int getVoiceChangerFactor ()
- nexAudioEdit getAudioEdit ()
- byte[] getVideoUUID ()
- void setVignetteEffect (boolean on)
- boolean getVignetteEffect ()

Static Public Member Functions

- static void setThumbTempDir (String path)
- static nexClip dup (nexClip src)
- static nexClip getSupportedClip (String path)
- static nexClip getSupportedClip (String path, boolean useCache)
- static void setVideoClipDetailThumbnailsDiskLimit (long size)
- static boolean cancelThumbnails ()
- static nexClip getSolidClip (int argbColor)

Static Public Attributes

- static final int AVC_Profile_Unknown = 0
- static final int AVC_Profile_Baseline = 66
- static final int AVC_Profile_Main = 77
- static final int AVC_Profile_Extended = 88
- static final int AVC_Profile_High = 100
- static final int AVC_Profile_High10 = 100

- static final int AVC_Profile_High422 = 122
- static final int AVC_Profile_High444 = 244
- static final int kClip_Rotate_0 = 0
- static final int kClip_Rotate_90 = 90
- static final int kClip_Rotate_180 = 180
- static final int kClip_Rotate_270 = 270
- static final int kCLIP_Supported = 0
- static final int kClip_Supported_Unknown = 1
- static final int kClip_NotSupported_AudioCodec = 2
- static final int kClip_NotSupported_AudioProfile = 3
- static final int kClip_NotSupported_Container = 4
- static final int kClip_NotSupported_ResolutionTooHigh = 5
- static final int kClip_NotSupported_DurationTooShort = 6
- static final int kClip_NotSupported_ResolutionTooLow = 7
- static final int kClip_NotSupported_VideoProfile = 8
- static final int kClip_NotSupported_VideoCodec = 9
- static final int kClip_NotSupported_VideoFPS = 10
- static final int kClip_NotSupported_VideoLevel = 11
- static final int kClip_NotSupported = 12
- static final int kClip_Supported_NeedResolutionTranscoding = 13
- static final int kClip_Supported_NeedFPSTranscoding = 14
- static final int kCLIP_TYPE_NONE = 0x00000000
- static final int kCLIP_TYPE_IMAGE = 0x00000001
- static final int kCLIP_TYPE_AUDIO = 0x00000003
- static final int kCLIP_TYPE_VIDEO = 0x00000004
- static final int **kCLIP_VIDEORENDERMODE_NORMAL** = 0
- static final int **kCLIP_VIDEORENDERMODE_360VIDE** = 1

Protected Member Functions

- int **getCombinedBrightness** ()
- int **getCombinedContrast** ()
- int **getCombinedSaturation** ()
- int **getTintColor** ()
- int **getLUTId** ()
- **nexClip** (nexClip src)
- void **setProjectUpdateSignal** (boolean onlyLoadList)
- final boolean **getAttachmentState** ()
- final void **setAttachmentState** (boolean set, nexObserver project)
- int **runDuration** ()
- int **getProjectDuration** ()

Static Protected Member Functions

- static nexClip **clone** (nexClip src)

Protected Attributes

- int **mTitleEffectStartTime**
- int **mTitleEffectEndTime**
- int **mStartTime** = 0
- int **mEndTime** = 0
- int **mDuration** = nexProject.kAutoThemeClipDuration

4.30.1 Detailed Description

This class manages image clips, video clips and audio clips to be edited with the NexEditor™ SDK. NexEditor™ S-DK supports the following file formats:

- Supported video file formats : MP4, 3GP and MOV.
- Supported video codec : H.264
- Supported audio file formats : MP3 and AAC.
- Supported image file formats : Android dependency.

The clips will be saved in the path of the original media content which the clip was created from and if the format of the clip is unsupported, kCLIP_TYPE_NONE will be returned from getClipType(). Another reason for a clip to be unsupported may be because of the resolution limitations from the hardware performance. Only after a clip is added to com.nexstreaming.nexeditorsdk.nexProject can it be edited or have effects applied. When editing a clip that is not added to `nexProject`, `ProjectNotAttachedException` will occur.

Example code :

```
nexClip clip = nexClip.getSupportedClip("/mnt/sdcard/1.jpg");
    if( clip ){
        if( clip.getClipType() == nexClip.kCLIP_TYPE_IMAGE ){
            Log.i(Tag,"wid = "+ nexClip.getWidth());
        }else if( clip.getClipType() == nexClip.kCLIP_TYPE_VIDEO ){
            Log.i(Tag,"time = "+ nexClip.getTotalTime());
        }else if( clip.getClipType() == nexClip.kCLIP_TYPE_NONE ){
            Log.e(Tag,"not supported clip = "+ clip.getSupportedResult());
        }
    }
```

Since

version 1.0.0

See Also

com.nexstreaming.nexeditorsdk.nexProject# add(int, boolean, nexClip)
com.nexstreaming.nexeditorsdk.nexProject::getClip(int, boolean)

4.30.2 Constructor & Destructor Documentation

4.30.2.1 com.nexstreaming.nexeditorsdk.nexClip.nexClip (String *path*)

This method creates a clip as an instance of `nexClip` and gets the media information about that clip. Therefore, it may take a while to create a particular `nexClip`, depending on the size and format of the file.

Example code:

class sample code

```
mClip = new nexClip(m_listfilepath.get(0));
```

Parameters

path	The location where the media clip is saved.

See Also

getSupportedClip(String)
getSupportedResult()

Since

version 1.0.0

4.30.2.2 com.nexstreaming.nexeditorsdk.nexClip.nexClip (String *path,* boolean *useCache*)

This method creates a clip as an instance of `nexClip` and gets the media information about that clip. Therefore, it may take a while to create a particular `nexClip`, depending on the size and format of the file.

This method allows to decide whether to cache retrieved clip's media information. Once a clip's media information is cached, this method will operate faster when the same clip is called again. It is recommended to set the parameter `useCache` to `FALSE` when a clip is for temporary use (for example, when a panorama image is made with number of trailing images, these images will be deleted once a new panorama image is made).

Example code:

```
mClip = new nexClip(m_listfilepath.get(0), true);
```

Parameters

path	The location where the media clip is saved.
useCache	Set to `TRUE` to cache retrieved media information; otherwise set to `FALSE`.

Since

version 1.3.43

4.30.3 Member Function Documentation

4.30.3.1 static boolean com.nexstreaming.nexeditorsdk.nexClip.cancelThumbnails ( ) `[static]`

This method cancels the thumbnail retrieving process.

Returns

`TRUE` if successful, otherwise `FALSE`.

Since

version 1.3.14

Deprecated For internal use only. Please do not use.

4.30.3.2 static nexClip com.nexstreaming.nexeditorsdk.nexClip.dup (nexClip *src*) [static]

This method creates a new nexClip by duplicating an existing nexClip passed with this method.

When a nexClip is duplicated, only the content information will be duplicated with default settings. Effects or other property settings from the original nexClip, that the new nexClip is being copied from, will not be duplicated.

Example code :

```
nexClip c = project.getClip(0,true);
    for(int i = 0 ; i < 3 ; i++){
        project.add(nexClip.dup(c));
    }
```

Parameters

src	An existing instance of nexClip to be duplicated.

Returns

The new instance of the nexClip clip.

Since

version 1.0.0

4.30.3.3 int com.nexstreaming.nexeditorsdk.nexClip.getAudioBitrate ( )

This method gets the average bitrate value of the audio file of a clip. The average bitrate value should follow what's specified on the header file of the audio file even if the specified value is 0.

Example code:

```
mAudioBitrate = mClip.getAudioBitrate();
        Log.d("ClipInfo", "Video bitrate ="+mVideoBitrate + "Audio bitrate ="+mAudioBitrate);
```

Returns

Average bitrate value if the media content is an audio file; otherwise 0.

Since

version 1.3.43

4.30.3.4 int com.nexstreaming.nexeditorsdk.nexClip.getAudioDuration ( )

This method gets the duration of an audio file in msec (milliseconds.) The duration value should follow what's specified on the header file of the audio file even if the specified value is 0.

Example code:

```
mAudioDuration = mClip.getAudioDuration();
        Log.d("ClipInfo", "Video duration ="+mVideoDuration + "Audio duration ="+mAudioDuration);
```

Returns

The duration of an audio file in `msec` (milliseconds)

Since

version 1.3.43

4.30.3.5 nexAudioEdit com.nexstreaming.nexeditorsdk.nexClip.getAudioEdit ( )

Returns

The audio edit class of the clip.

Since

1.5.43

4.30.3.6 nexAudioEnvelop com.nexstreaming.nexeditorsdk.nexClip.getAudioEnvelop ( )

This method gets an audio item to add or adjust the audio envelop.

Example code:

```
for(int i=0; i<mTime.size(); i++){
    project.getClip(0,false).getAudioEnvelop().addVolumeEnvelope(Integer.parseInt(mTime.get(
      i)),Integer.parseInt(mVol.get(i)));
}
```

Returns

nexAudioEnvelop An audio item.

Since

version 1.3.52

4.30.3.7 boolean com.nexstreaming.nexeditorsdk.nexClip.getAudioOnOff ( )

This method checks whether a clip's audio is on or off.

Example code:

```
visualClip.mAudioOnOff = item.getAudioOnOff()?1:0;
```

Returns

`TRUE` if the audio is on; otherwise `FALSE` if the audio is off.

See Also

setAudioOnOff(boolean)

Since

version 1.0.0

4.30.3.8 int com.nexstreaming.nexeditorsdk.nexClip.getAVCLevel ( )

This method gets the level of a video file if the video file format is H.264.

Example code:

```
mSupportedresult.setText("AVC Profile:"+mClip.getAVCProfile() + ", Level:"+mClip.getAVCLevel());
```

Returns

The level of the video file.

Since

version 1.3.4

4.30.3.9 int com.nexstreaming.nexeditorsdk.nexClip.getAVCProfile ( )

This method gets the profile value of a video file if the video file format is H.264.

Example code:

```
mSupportedresult.setText("AVC Profile:"+mClip.getAVCProfile() + ", Level:"+mClip.getAVCLevel());
```

Returns

The profile value of the video file.

Since

version 1.3.4

4.30.3.10 int com.nexstreaming.nexeditorsdk.nexClip.getBGMVolume ( )

This method gets the volume of a project's background music (BGM).

Example code:

```
visualClip.mBGMVolume = clipItem.getBGMVolume();
```

Returns

The volume of BGM, as an `integer`. This value will be in the range of 0 to 200.

See Also

setBGMVolume(int)

Since

version 1.1.0

4.30.3.11 int com.nexstreaming.nexeditorsdk.nexClip.getBrightness ( )

This method gets the brightness of a clip.

Example code:

```
mBrightness = clip.getBrightness();
```

Returns

The brightness set to the clip, in the range of -255 to 255.

See Also

setBrightness(int)

Since

version 1.1.0

4.30.3.12 nexClipEffect com.nexstreaming.nexeditorsdk.nexClip.getClipEffect ( )

This method gets an instance of a clip effect to set on a clip that is currently added to a project. If this method is called and there are currently no clips added to a project, `ProjectNotAttachedException` will occur.

Example code:

```
mProject.getClip(i, true).getClipEffect().setEffectShowTime(0,clip.getTotalTime()); //set full
       time effect
```

Returns

The `nexClipEffect` instance.

Exceptions

ProjectNotAttached-Exception	

Since

version 1.0.0

4.30.3.13 nexClipEffect com.nexstreaming.nexeditorsdk.nexClip.getClipEffect (boolean *internal*)

This method gets the clip effect

Parameters

internal	Set to `TRUE` when using internally.

Returns

clip effect

Since

version 1.5.15

4.30.3.14 int com.nexstreaming.nexeditorsdk.nexClip.getClipType ( )

This method gets the type of a clip.

Example code:

```
if( mClip.getClipType() == nexClip.kCLIP_TYPE_VIDEO ) {
               mClip.loadVideoClipThumbnails( new nexClip.OnLoadVideoClipThumbnailListener(){
               @Override
                   public void onLoadThumbnailResult(int event) {}
               }
           }
```

Returns

One of:

- kCLIP_TYPE_VIDEO = { kCLIP_TYPE_VIDEO} : Video
- kCLIP_TYPE_AUDIO = { kCLIP_TYPE_AUDIO} : Audio
- kCLIP_TYPE_IMAGE = { kCLIP_TYPE_IMAGE} : Image
- kCLIP_TYPE_NONE = { kCLIP_TYPE_NONE} : None

Since

version 1.0.0

4.30.3.15 int com.nexstreaming.nexeditorsdk.nexClip.getClipVolume ( )

This method gets the volume of a clip.

Example code:

```
item.mClipVolume = aClip.getClipVolume();
```

Returns

The volume of the clip, as an `integer`. This value will be in the range of 0 to 200 and the default value is 100.

See Also

setClipVolume(int)

Since

version 1.1.0

4.30.3.16 nexColorEffect com.nexstreaming.nexeditorsdk.nexClip.getColorEffect ( )

This method gets the value of a color effect on a clip.

Example code:

```
if(color_filter != -1) {
    clip2.setColorEffect(getColorEffect(color_filter - 1));
}
```

Returns

The value of the color effect set on the clip.

See Also

setColorEffect(nexColorEffect)

Since

version 1.1.0

4.30.3.17 int com.nexstreaming.nexeditorsdk.nexClip.getContrast ( )

This method gets the contrast of a clip.

Example code:

```
mContrast = clip.getContrast();
```

Returns

The contrast set to the clip, in the range of -255 to 255.

See Also

setContrast(int)

Since

version 1.1.0

4.30.3.18 nexCrop com.nexstreaming.nexeditorsdk.nexClip.getCrop ( )

This method crops a clip.

Example code:

```
for( int i = 0 ; i < project.getTotalClipCount(true) ; i++ ){
            project.getClip(i, true).getCrop().randomizeStartEndPosition(false,
      nexCrop.CropMode.PAN_RAND);
        }
```

Returns

A cropped section of the clip, as an instance of `nexCrop`.

See Also

nexCrop

Since

version 1.1.0

4.30.3.19 int com.nexstreaming.nexeditorsdk.nexClip.getFramesPerSecond ( )

This method gets the FPS (Frames Per Second) value of a clip.

Example code:

```
mFps = (TextView)findViewById(R.id.textView5);
    mFps.setText("Clip FPS : "+mClip.getFramesPerSecond());
```

Returns

The value of FPS, as an `integer` .

Since

version 1.1.0

4.30.3.20 int com.nexstreaming.nexeditorsdk.nexClip.getHeight ( )

This method gets the horizontal resolution information if the clip is either an image clip or a video clip.

Example code:

```
mResolution.setText("Width="+mClip.getWidth()+", Height="+mClip.getHeight());
```

}

Returns

The horizontal resolution of the clip, in `pixels`.

See Also

getWidth()

Since

version 1.0.0

4.30.3.21 int com.nexstreaming.nexeditorsdk.nexClip.getImageClipDuration ( )

This method gets the duration of an image clip to be played within the project.

Example code:

```
visualClip.mTotalTime = clipItem.getImageClipDuration();
```

Returns

The duration of an image clip in `msec` (milliseconds).

See Also

setImageClipDuration(int)

Since

version 1.0.0

4.30.3.22 Bitmap com.nexstreaming.nexeditorsdk.nexClip.getMainThumbnail (float *height,* float *scaleDIPtoPX*)

This method gets the main thumbnail image of a clip. Run `loadVideoClipThumbnails()` to get a thumbnail image from a video clip.

To use this API to get a video clip thumbnail image, `loadVideoClipThumbnails()` should be called successfully first, which is indicated by the event: loadVideoClipThumbnails(com.nexstreaming.nexeditorsdk.nexClip.-OnLoadVideoClipThumbnailListener)

Example code :

```
mClip = new nexClip(m_listfilepath.get(0));
   Bitmap bm = mClip.getMainThumbnail(240f ,getBaseContext().getResources().getDisplayMetrics().density);
   mIV = (ImageView)findViewById(R.id.imageView);
   mIV.setImageBitmap(bm);
```

Parameters

height	The height of the thumbnail image.
scaleDIPtoPX	The value of DIP.

Returns

The thumbnail image for the clip as a bitmap.

See Also

getVideoClipTimeLineOfThumbnail()
setMainThumbnail(android.graphics.Bitmap)
loadVideoClipThumbnails(com.nexstreaming.nexeditorsdk.nexClip.OnLoadVideoClipThumbnailListener)

Since

version 1.0.0

4.30.3.23 String com.nexstreaming.nexeditorsdk.nexClip.getPath ( )

This method gets the path of the current clip.

Example code:

```
if (project.getClip(j, false).getPath() != clip.getPath())
```

Returns

The location of the clip saved.

Since

version 1.0.0

4.30.3.24 int com.nexstreaming.nexeditorsdk.nexClip.getProjectEndTime ( )

This method gets the end time of a clip when playing a project. This can be considered the location of a clip in a project's timeline.

Example code:

```
int end = mProject.getClip(i, true).getProjectEndTime();
```

Returns

The clip end time in `msec` (milliseconds).

See Also

getProjectStartTime()

Since

version 1.0.0

4.30.3.25 int com.nexstreaming.nexeditorsdk.nexClip.getProjectStartTime ( )

This method gets the start time of a clip when playing a project. This can be considered the location of a clip in a project's timeline.

Example code:

```
if (project.getClip(i, false).getProjectStartTime() <timeline1_endtime) {
    gototimeline2 = true;
    break;
}
```

Returns

The clip start time in `msec` (milliseconds).

See Also

getProjectEndTime()

Since

version 1.0.0

4.30.3.26 int com.nexstreaming.nexeditorsdk.nexClip.getRotateDegree ( )

This method gets the degrees of rotation set on a clip, rotated counterclockwise, if the clip is an image or video clip.

Example code:

```
rt.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                mEngine.getProject().getClip(0, true).setRotateDegree(mEngine.getProject().getClip(0, true)
      .getRotateDegree() + 90);
                setDimension();
            }
        });
```

Returns

The number of degrees the clip has been rotated counterclockwise. This will be one of the following:

- kClip_Rotate_0 = { kClip_Rotate_0}
- kClip_Rotate_90 = { kClip_Rotate_90}
- kClip_Rotate_180 = { kClip_Rotate_180}
- kClip_Rotate_270 = { kClip_Rotate_270}

See Also

setRotateDegree(int)

Since

version 1.0.1

4.30.3.27 int com.nexstreaming.nexeditorsdk.nexClip.getRotateInMeta ( )

This method gets the orient angle of the metadata of a file if the file format is MP4.

Example code:

```
mProject.getClip(i, true).setRotateDegree(clip.getRotateInMeta());
```

Returns

The orient angle of the metadata.

Since

version 1.3.4

4.30.3.28 int com.nexstreaming.nexeditorsdk.nexClip.getSaturation ( )

This method gets the color saturation of a clip.

Example code:

```
mSaturation = clip.getSaturation();
```

Returns

The color saturation set on the clip, in the range of -255 to 255.

See Also

setSaturation(int)

Since

version 1.1.0

4.30.3.29 int com.nexstreaming.nexeditorsdk.nexClip.getSeekPointCount ( )

This method finds out the number of I-frames of the media content.

Returns

The number of IFrames.

See Also

getSeekPointInterval()

Since

1.5.42

4.30.3.30 int com.nexstreaming.nexeditorsdk.nexClip.getSeekPointInterval ( )

This method finds out the average interval of IFrames of the media content.

Returns

The average interval of I-frames in `msec` (milliseconds).

See Also

getSeekPointCount()

Since

1.5.42

4.30.3.31 int [ ] com.nexstreaming.nexeditorsdk.nexClip.getSeekPointsSync ( )

This method gets the location of the seek table (including I-frame and IDR) if the file is a video file.

Example code:

```
int tab[] = mClip.getSeekPointsSync();
        seekTab = new int[2];
        if( tab.length > 2 ){
            seekTab[0] = tab[1];
            Log.d(TAG,"user tab time1 ="+seekTab[0]);
            seekTab[1] = tab[tab.length-1];
            Log.d(TAG,"user tab time2 ="+seekTab[1]);
            startTime = 0;
            endTime= 0;
            maxCount = seekTab.length;
        }
```

Returns

The CTS value of the I-frame as an `array` .

See Also

com.nexstreaming.nexeditorsdk.nexEngine::getIDRSeekTabSync(nexClip)

Since

version 1.3.4

4.30.3.32 static nexClip com.nexstreaming.nexeditorsdk.nexClip.getSolidClip (int *argbColor*) [static]

This method creates a clip with a solid color specified to the parameter `argbColor`. The clip will be made independently by the NexEditor™ SDK without using any external media content, in the purpose of making a background color.

Example code:

```
nexClip clip = nexClip.getSolidClip(Color.RED);
```

Parameters

argbColor	An ARGB value of a color to set to a clip as `integer`.

Returns

A clip with a solid background color.

See Also

getSolidColor()
setSolidColor(int)

Since

version 1.3.43

4.30.3.33 int com.nexstreaming.nexeditorsdk.nexClip.getSolidColor ( )

This method gets the ARGB value of the color set to a clip if the clip was made by the method `getSolidClip()`.

Example code:

```
int color = clip.getSolidColor();
```

Returns

The ARGB value of the color if the clip is made by `getSolidClip()`; otherwise 0.

See Also

setSolidColor(int)

Since

version 1.3.43

4.30.3.34 static nexClip com.nexstreaming.nexeditorsdk.nexClip.getSupportedClip (String *path*) [static]

This method only creates a media clip that is supported by the NexEditor™ SDK.

Example code :

```
nexClip clip = nexClip.getSupportedClip("/mnt/sdcard/1.jpg");
    if( clip == null ){
        Log.e(Tag,"not supported clip");
    }
```

Parameters

path	The location where the media clip is saved.

Returns

A `nexClip` instance if the media clip is supported by the NexEditor™ SDK.

See Also

nexClip(String)

Since

version 1.0.0

4.30.3.35 static nexClip com.nexstreaming.nexeditorsdk.nexClip.getSupportedClip (String *path*, boolean *useCache*) [static]

This method only creates a media clip that is supported by the NexEditor™ SDK.

Example code:

```
nexClip clip2= nexClip.getSupportedClip(m_soundfilepath.get(0));
```

Parameters

path	The location where the media clip is saved.
useCache	Set to TRUE to cache retrieved media information; otherwise set to FALSE. This method allows to decide whether to cache retrieved clip's media information. Once a clip's media information is cached, this method will operate faster when the same clip is called again. It is recommended to set the parameter useCache to FALSE when a clip is for temporary use (for example, when a panorama image is made with number of trailing images, these images will be deleted once a new panorama image is made).

Returns

A nexClip instance if the media clip is supported by the NexEditor™ SDK.

Since

version 1.3.43

4.30.3.36 int com.nexstreaming.nexeditorsdk.nexClip.getSupportedResult ( )

This methods checks whether a clip is supported by the NexEditor™ SDK.

Example code:

```
switch (mClip.getSupportedResult())
        {
            case 0 : //mClip.kCLIP_Supported :
            mSupportedresult.setText("Clip is supported");
            break;
        }
```

Returns

One of:

- kClip_NotSupported = { kClip_NotSupported}
- kClip_NotSupported_AudioCodec = { kClip_NotSupported_AudioCodec}
- kClip_NotSupported_AudioProfile = { kClip_NotSupported_AudioProfile}
- kClip_NotSupported_Container = { kClip_NotSupported_Container}
- kClip_NotSupported_DurationTooShort = { kClip_NotSupported_DurationTooShort}
- kClip_NotSupported_ResolutionTooHigh = { kClip_NotSupported_ResolutionTooHigh}
- kClip_NotSupported_ResolutionTooLow = { kClip_NotSupported_ResolutionTooLow}
- kClip_NotSupported_VideoCodec = { kClip_NotSupported_VideoCodec}

- kClip_NotSupported_VideoFPS = { kClip_NotSupported_VideoFPS}
- kClip_NotSupported_VideoLevel = { kClip_NotSupported_VideoLevel}
- kClip_NotSupported_VideoProfile = { kClip_NotSupported_VideoProfile}
- kClip_Supported_NeedFPSTranscoding = { kClip_Supported_NeedFPSTranscoding}
- kClip_Supported_NeedResolutionTranscoding = { kClip_Supported_NeedResolutionTranscoding}
- kCLIP_Supported = { kCLIP_Supported}
- kClip_Supported_Unknown = { kClip_Supported_Unknown}

Since

version 1.0.0

4.30.3.37 int com.nexstreaming.nexeditorsdk.nexClip.getTotalTime ( )

This method gets the playback time if the clip is either an audio clip or a video clip.

Example code:

```
if( clip != null ){
        clip.setImageClipDuration(clip2.getTotalTime());
    }
```

Returns

Playback time of the clip in `msec` (milliseconds).

Since

version 1.0.0

4.30.3.38 nexTransitionEffect com.nexstreaming.nexeditorsdk.nexClip.getTransitionEffect ( )

This method gets an instance of a transition effect to set on a clip that is currently added to a project.

If this method is called and there are currently no clips added to a project, `ProjectNotAttachedException` will occur.

Example code:

```
clip.getTransitionEffect().setTransitionEffect("com.nexstreaming.unkownid");
```

Returns

The `nexTransitionEffect` instance.

Exceptions

ProjectNotAttached-Exception	

Since

version 1.0.0

4.30.3.39 nexTransitionEffect com.nexstreaming.nexeditorsdk.nexClip.getTransitionEffect (boolean *internal*)

This method gets the transition effect

Parameters

internal	Set to `TRUE` when using internally.

Returns

transition effect

Since

version 1.5.15

4.30.3.40 int com.nexstreaming.nexeditorsdk.nexClip.getVideoBitrate ( )

This method gets the average bitrate value of the video file of a clip. The average bitrate value should follow what's specified on the header file of the video file even if the specified value is 0.

Example code:

```
mEngine.export(mFile.getAbsolutePath(), clip.getWidth(), clip.getHeight(), clip.getVideoBitrate(),
      Long.MAX_VALUE, rotateInMeta);
```

Returns

Average bitrate value if the media content is a video file; otherwise 0.

Since

version 1.3.43

4.30.3.41 int com.nexstreaming.nexeditorsdk.nexClip.getVideoClipDetailThumbnails (int *thumbWidth,* int *thumbHeight,* final int *startTime,* final int *endTime,* int *maxCount,* int *rotate,* final OnGetVideoClipDetailThumbnailsListener *listener*)

This method gets the thumbnails from I-frames within the specific section of the clip. This method does not cache retrieved thumbnails, therefore the developer has to cache it separately if needed.

Note

The bitmaps of `onGetDetailThumbnailResult()` returned from the listener operates in singleton, therefore it should be saved separately to avoid overwrite with other bitmaps. If both `startTime` and `endTime` value is set to 0, only I-frames will be retrieved.

Example code:

```
project.getClip(0, true).getVideoClipDetailThumbnails(project.getClip(0, true)
      .getWidth(), project.getClip(0, true).getHeight(), 0, thumbendtime, 100, 0, new
      nexClip.OnGetVideoClipDetailThumbnailsListener() {}
```

Parameters

thumbWidth	The width resolution of the thumbnail as integer.
thumbHeight	The height resolution of the thumbnail as integer.
startTime	The start time of the specific section of the clip to retrieve thumbnails.
endTime	The end time of the specific section of the clip to retrieve thumbnails.
maxCount	Maximum number of thumbnails to get.
rotate	The rotation angle of the thumbnail as integer (0 , 90 , 180 , 270).
listener	The instance of `OnGetVideoClipDetailThumbnailsListener`.

Returns

-1 if there is no listener or the clip is not a video clip; otherwise 0.

See Also

getVideoClipDetailThumbnails(int, int, int, int, int, int, boolean, int[], com.nexstreaming.nexeditorsdk.nexClip.OnGetVideoClipDetailThumbnailsListener)
getVideoClipDetailThumbnails(int, int, int[], int, com.nexstreaming.nexeditorsdk.nexClip.OnGetVideoClipDetailThumbnailsListener)

Since

version 1.3.0

4.30.3.42 int com.nexstreaming.nexeditorsdk.nexClip.getVideoClipDetailThumbnails (int *thumbWidth,* int *thumbHeight,* int[ ] *arrayTimeTab,* int *rotate,* final OnGetVideoClipDetailThumbnailsListener *listener*)

This method gets the thumbnails from I-frames within the specific section of the clip. This method does not cache retrieved thumbnails, therefore the developer has to cache it separately if needed.

Example code:

```
project.getClip(0, true).getVideoClipDetailThumbnails(project.getClip(0, true)
      .getWidth(), project.getClip(0, true).getHeight(), seekTab, 0, new nexClip.
      OnGetVideoClipDetailThumbnailsListener() {}
```

Parameters

thumbWidth	The width resolution of the thumbnail as integer.
thumbHeight	The height resolution of the thumbnail as integer.
arrayTimeTab	An array of thumbnails within the specific section of the clip as integer. The thumbnails must be retrieved only from I-frames which can be found with `getSeekPointsSync()`.
rotate	The rotation angle of the thumbnail as integer (0 , 90 , 180 , 270).
listener	The instance of `OnGetVideoClipDetailThumbnailsListener`.

Returns

-1 if there is no listener or the clip is not a video clip; otherwise 0.

See Also

getVideoClipDetailThumbnails(int, int, int, int, int, int, boolean, int[], com.nexstreaming.nexeditorsdk.nexClip.OnGetVideoClipDetailThumbnailsListener)
getVideoClipDetailThumbnails(int, int, int, int, int, int, com.nexstreaming.nexeditorsdk.nexClip.OnGetVideoClipDetailThumbnailsListener)

Since

version 1.3.43

4.30.3.43 int com.nexstreaming.nexeditorsdk.nexClip.getVideoClipDetailThumbnails (int *thumbWidth,* int *thumbHeight,* final int *startTime,* final int *endTime,* int *maxCount,* int *rotate,* boolean *noCache,* int[ ] *arrayTimeTab,* final OnGetVideoClipDetailThumbnailsListener *listener*)

This method gets the thumbnails from I-frames within the specific section of the clip. This method does not cache retrieved thumbnails, therefore the developer has to cache it separately if needed. This method allows to set all the options available for retrieving thumbnails.

Example code:

```
mClip.getVideoClipDetailThumbnails(mWidth, mHeight, startTime, endTime, maxCount, mAngle, noCache, seekTab,
      new nexClip.OnGetVideoClipDetailThumbnailsListener() {}
```

Parameters

thumbWidth	The width resolution of the thumbnail as integer.
thumbHeight	The height resolution of the thumbnail as integer.
startTime	The start time of the specific section of the clip to retrieve thumbnails.
endTime	The end time of the specific section of the clip to retrieve thumbnails.
maxCount	Maximum number of thumbnails to get.
rotate	The rotation angle of the thumbnail as integer (0 , 90 , 180 , 270).
noCache	Set to `TRUE` to retrieve thumbnails without caching them; otherwise `FALSE`.
arrayTimeTab	An array of thumbnails within the specific section of the clip as integer. The thumbnails must be retrieved only from I-frames which can be found with `getSeekPointsSync()`.
listener	The instance of `OnGetVideoClipDetailThumbnailsListener`.

Returns

-1 if there is no listener or the clip is not a video clip; otherwise 0.

See Also

getVideoClipDetailThumbnails(int, int, int, int, int, int, com.nexstreaming.nexeditorsdk.nexClip.OnGetVideoClipDetailThumbnailsListener)
getVideoClipDetailThumbnails(int, int, int[], int, com.nexstreaming.nexeditorsdk.nexClip.OnGetVideoClipDetailThumbnailsListener)

Since

version 1.3.43

4.30.3.44 nexVideoClipEdit com.nexstreaming.nexeditorsdk.nexClip.getVideoClipEdit ( )

This method gets the editing related properties for a clip, except for trimming or speed control, if the clip is a video clip.

Example code:

```
mProject.getClip(0, true).getVideoClipEdit().setTrim(0, trim);
```

Returns

The `nexVideoClipEdit` that has been set for the clip.

Exceptions

ProjectNotAttachedException	

Since

version 1.0.0

4.30.3.45 int com.nexstreaming.nexeditorsdk.nexClip.getVideoClipIDR2YOnlyThumbnails (int *thumbWidth,* int *thumbHeight,* final int *startTime,* final int *endTime,* int *maxCount,* final OnGetVideoClipIDR2YOnlyThumbnailsListener *listener*)

Deprecated For internal use only. Please do not use.

4.30.3.46 int com.nexstreaming.nexeditorsdk.nexClip.getVideoClipIFrameThumbnails (int *thumbWidth,* int *thumbHeight,* final OnGetVideoClipDetailThumbnailsListener *listener*)

This method gets the thumbnails(maximum of 3600 thumbnails) from all the I-frames of a clip. This method does not cache retrieved thumbnails, therefore the developer has to cache it separately if needed.

Note

: The bitmaps of `onGetDetailThumbnailResult()` returned from the listener operates in singleton, therefore it should be saved separately to avoid overwrite with other bitmaps.

Parameters

thumbWidth	The width resolution of the thumbnail.
thumbHeight	The height resolution of the thumbnail.
listener	The instance of `OnGetVideoClipDetailThumbnailsListener`.

Returns

-1 if there is no listener or the clip is not a video clip; otherwise 0.

See Also

getVideoClipDetailThumbnails(int, int, int, int, int, int, boolean, int[], com.nexstreaming.nexeditorsdk.nexClip.OnGetVideoClipDetailThumbnailsListener)
getVideoClipDetailThumbnails(int, int, int, int, int, int, com.nexstreaming.nexeditorsdk.nexClip.OnGetVideoClipDetailThumbnailsListener)
getVideoClipDetailThumbnails(int, int, int[], int, com.nexstreaming.nexeditorsdk.nexClip.OnGetVideoClipDetailThumbnailsListener)

Since

version 1.3.0

Deprecated For internal use only. Please do not use.

4.30.3.47 int [ ] com.nexstreaming.nexeditorsdk.nexClip.getVideoClipTimeLineOfThumbnail ( )

This method gets a thumbnail image's extracted time in sequence.

To use this API, `loadVideoClipThumbnails()` should first be called successfully, which is indicated by the event: loadVideoClipThumbnails(com.nexstreaming.nexeditorsdk.nexClip.OnLoadVideoClipThumbnailListener) In the case that the clip is a video, a maximum of 50 thumbnail images can be saved.

Returns

The extracted time of a thumbnail image in sequence as an `integer`, in `msec` (milliseconds).

Exceptions

ClipIsNotVideoException	

Example code :

```
int timeLine[] = mClip.getVideoClipTimeLineOfThumbnail();
    for( int i = 0 ; i < timeLine.length ; i++){
        Bitmap bm2 = mClip.getVideoClipTimeLineThumbnail(0,timeLine[i],false,false);
        mIV.setImageBitmap(bm2);
    }
```

See Also

getVideoClipTimeLineThumbnail(int, int, boolean, boolean)
loadVideoClipThumbnails(com.nexstreaming.nexeditorsdk.nexClip.OnLoadVideoClipThumbnailListener)

Since

version 1.0.0

4.30.3.48 Bitmap com.nexstreaming.nexeditorsdk.nexClip.getVideoClipTimeLineThumbnail (int *rotation,* int *time,* boolean *fliph,* boolean *flipv*)

This method gets a thumbnail image, that's been extracted from a video clip, matching the input parameter value.

To use this API, `loadVideoClipThumbnails()` should be called successfully first, which is indicated by the event: loadVideoClipThumbnails(com.nexstreaming.nexeditorsdk.nexClip.OnLoadVideoClipThumbnailListener) In the case of a video clip, a maximum of 50 thumbnails can be saved. One thumbnail image will be extracted depending on the parameter values input.

Parameters

rotation	Input the rotation value of a thumbnail to be printed, in degrees. (only values accepted: 0, 90, 180, 270)
time	The duration of a thumbnail to get.
fliph	Set to `TRUE` to make a thumbnail based on the horizontal resolution.
flipv	Set to `TRUE` to make a thumbnail based on the vertical resolution.

Returns

Bitmap of a thumbnail.

Since

version 1.0.0

Exceptions

ClipIsNotVideoException	

Example code :

```
int timeLine[] = mClip.getVideoClipTimeLineOfThumbnail();
    for( int i = 0 ; i < timeLine.length ; i++){
        Bitmap bm2 = mClip.getVideoClipTimeLineThumbnail(0,timeLine[i],false,false);
        mIV.setImageBitmap(bm2);
    }
```

See Also

getVideoClipTimeLineOfThumbnail()
#loadVideoClipThumbnails(OnLoadVideoClipThumbnailListener)

Since

version 1.0.0

4.30.3.49 int com.nexstreaming.nexeditorsdk.nexClip.getVideoDuration ( )

This method gets the duration of a video in `msec` (milliseconds). The duration value should follow what's specified on the header file of the video file even if the specified value is 0.

Example code:

```
mVideoDuration = mClip.getVideoDuration();
        Log.d("ClipInfo", "Video duration ="+mVideoDuration + "Audio duration ="+mAudioDuration);
```

Returns

The duration of a video file in `msec` (milliseconds).

Since

version 1.3.43

4.30.3.50 byte [ ] com.nexstreaming.nexeditorsdk.nexClip.getVideoUUID ( )

Returns

Since

1.5.43

4.30.3.51 boolean com.nexstreaming.nexeditorsdk.nexClip.getVignetteEffect ( )

Returns

Since

1.5.43

4.30.3.52 int com.nexstreaming.nexeditorsdk.nexClip.getVoiceChangerFactor ( )

This method gets the voice changer factor set to a clip.

Returns

The voice changer factor, as an `integer` (0 ∼ 4).

Since

version 1.5.15

See Also

setVoiceChangerFactor(int)

4.30.3.53 int com.nexstreaming.nexeditorsdk.nexClip.getWidth ( )

This method gets the vertical resolution information if the clip is either an image clip or a video clip.

Example code:

```
mResolution.setText("Width="+mClip.getWidth()+", Height="+mClip.getHeight());
```

}

Returns

The vertical resolution of the clip, in `pixels`.

See Also

getHeight()

Since

version 1.0.0

4.30.3.54 boolean com.nexstreaming.nexeditorsdk.nexClip.hasAudio ( )

This method checks if there is any audio file included in the media content.

Example code:

class sample code

```
if(mClip.hasAudio() && mClip.hasVideo())
mExistaudiovideo.setText("Clip exist Audio and Video");
```

Returns

`TRUE` if there is any audio file, otherwise `FALSE`.

See Also

hasVideo()

Since

version 1.0.0

4.30.3.55 boolean com.nexstreaming.nexeditorsdk.nexClip.hasVideo ( )

This method checks if there is any video file included in the media content.

Example code:

```
if(mClip.hasAudio() && mClip.hasVideo())
        mExistaudiovideo.setText("Clip exist Audio and Video");
```

Returns

TRUE if there is any video file, otherwise FALSE.

See Also

hasAudio()

Since

version 1.0.0

4.30.3.56 boolean com.nexstreaming.nexeditorsdk.nexClip.isSolid ( )

This method indicates whether or not a clip created by getSolidClip(int) is solid.

Returns

TRUE if it is a solid clip; otherwise, FALSE if it is an image, video, or an audio clip.

See Also

getSolidClip(int)

Since

1.5.23

4.30.3.57 int com.nexstreaming.nexeditorsdk.nexClip.loadVideoClipThumbnails (final OnLoadVideoClipThumbnailListener *listener*)

This method gets a thumbnail image and caches it. Since this method operates asynchronously, its success can be checked with `OnGetThumbnailListener`. The thumbnail cannot be retrieved when `nexEngine` is in the `play` state.

If the clip is not a type of video, it returns -1.

Example code :

```
mClip = new nexClip(m_listfilepath.get(0));
mIV = (ImageView)findViewById(R.id.imageView);

if( mClip.getClipType() == nexClip.kCLIP_TYPE_VIDEO ) {
    mClip.loadVideoClipThumbnails( new nexClip.OnLoadVideoClipThumbnailListener(){
        @Override
        public void onLoadThumbnailResult(int event) {
            if( event == OnLoadVideoClipThumbnailListener.kEvent_Ok ) {
                Bitmap bm = mClip.getMainThumbnail(240f, getBaseContext().getResources().getDisplayMetrics(
      ).density);
                mIV.setImageBitmap(bm);
                int timeLine[] = mClip.getVideoClipTimeLineOfThumbnail();
                for( int i = 0 ; i < timeLine.length ; i++){
                    Bitmap bm2 = mClip.getVideoClipTimeLineThumbnail(0,timeLine[i],false,false);
                    mIV.setImageBitmap(bm2);
                }
            }
        }
    }
     );
}else{
    ....
}
```

Parameters

listener	OnLoadVideoClipThumbnailListener

Returns

-1 if the clip is not a video clip, otherwise 0.

See Also

getVideoClipTimeLineOfThumbnail()
getMainThumbnail(float, float)
getVideoClipTimeLineThumbnail(int, int, boolean, boolean)

Since

version 1.0.0

4.30.3.58 int com.nexstreaming.nexeditorsdk.nexClip.replaceClip (String *path*)

This method replaces clip's content with another content.

Example code:

```
if(project.getTotalClipCount(false) !=0) {
        project.getClip(0, false).replaceClip(m_soundfilepath);
    }
```

Parameters

path	The path of the content to replace the existing content of the clip.

Returns

0 if successful; -1 if the new content type doesn't match or if the new content is not supported.

Since

version 1.1.0

4.30.3.59 void com.nexstreaming.nexeditorsdk.nexClip.setAudioOnOff (boolean *on*)

This method can turn the audio on and off while a clip is being played within a project.

Example code:

```
clip.setAudioOnOff(false);
```

Parameters

on	Whether or not audio should be on. Set to `TRUE` to turn on the audio, otherwise set to `FALSE` to turn the audio off.

See Also

getAudioOnOff()

Since

version 1.0.0

4.30.3.60 boolean com.nexstreaming.nexeditorsdk.nexClip.setBGMVolume (int *volume*)

This method sets the volume of the background music (BGM) for a project.

Example code:

```
if( mBGMOnOff.isChecked() ) {
            mEngin.getProject().getClip(0, true).setBGMVolume(mBGMVol.getProgress());
        } else {
            mEngin.getProject().getClip(0, true).setBGMVolume(0);
        }
```

Parameters

volume	The volume of the BGM, as an `integer`. This value can range from 0 to 200 and the default value is 100.

Returns

`FALSE` if the input value is not in the range of 0 to 200, otherwise `TRUE`.

See Also

getBGMVolume()

Since

version 1.1.0

4.30.3.61 boolean com.nexstreaming.nexeditorsdk.nexClip.setBrightness (int *value*)

This method sets the brightness of a clip.

Example code:

```
project.getClip(1 + i, true).setBrightness(100);
```

Parameters

value	The brightness to set on the clip, as an `integer`. This must be in the range of -255 to 255.

Returns

`TRUE` if the value is set successfully; `FALSE` if the value is not in range.

See Also

getBrightness()

Since

version 1.1.0

4.30.3.62 boolean com.nexstreaming.nexeditorsdk.nexClip.setClipVolume (int *volume*)

This method sets the volume of a clip.

Example code:

```
project.getClip(0, false).setClipVolume(mBGMVol);
```

Parameters

volume	The volume of the clip, as an `integer`. This value can range from 0 to 200 and the default value is 100.

Returns

`FALSE` if the input value is not in the range of 0 to 200, otherwise `TRUE`.

See Also

getClipVolume()

Since

version 1.1.0

4.30.3.63 void com.nexstreaming.nexeditorsdk.nexClip.setColorEffect (nexColorEffect *effect*)

This method sets a color effect on a clip.

Parameters

effect	The color effect to set on the clip.

Example code :

```
clip.setColorEffect(nexColorEffect.ALIEN_INVASION);
```

Since

version 1.1.0

See Also

getColorEffect()

4.30.3.64 boolean com.nexstreaming.nexeditorsdk.nexClip.setContrast (int *value*)

This method sets the contrast of a clip.

Example code:

class sample code

```
project.getClip(1 + i, true).setContrast(-25);
```

Parameters

value	The contrast to set on the clip, as an `integer`. This must be in the range of -255 to 255.

Returns

`TRUE` if the value is set successfully; `FALSE` if the value is not in range.

See Also

getContrast()

Since

version 1.1.0

4.30.3.65 void com.nexstreaming.nexeditorsdk.nexClip.setImageClipDuration (int *ms*)

This method sets the duration of an image clip to be played within the project.

Example code:

```
clip.setImageClipDuration(clip2.getTotalTime());
```

Parameters

ms	The duration an image clip should be displayed, in `msec` (milliseconds).

See Also

getImageClipDuration()

Since

version 1.0.0

4.30.3.66 void com.nexstreaming.nexeditorsdk.nexClip.setMainThumbnail (Bitmap *thumb*)

This method registers a thumbnail image.

Example code:

```
bmlist.add(mClip.getMainThumbnail(240f ,getBaseContext().getResources().getDisplayMetrics().density));
        mClip.setMainThumbnail(bmlist.get(0));
```

Parameters

thumb	An instance of the thumbnail bitmap to be registered.

See Also

getMainThumbnail(float, float)

Since

version 1.0.0

4.30.3.67 void com.nexstreaming.nexeditorsdk.nexClip.setRotateDegree (int *degree*)

This method sets the degrees of rotation of a clip. A video file has its rotation degree information saved as metadata when it's being filmed. Use this method to rotate the video file before playing or exporting. As of a JPEG image, this method is not necessary because NexEditor™ SDK will automatically adjust the rotation degree.

Example code:

```
rt.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                mEngine.getProject().getClip(0, true).setRotateDegree(mEngine.getProject().getClip(0, true)
      .getRotateDegree() + 90);
                setDimension();
            }
        });
```

Parameters

degree	The number of degrees to set to a video clip as integer. This will be one of the following: • kClip_Rotate_0 = { kClip_Rotate_0} • kClip_Rotate_90 = { kClip_Rotate_90} • kClip_Rotate_180 = { kClip_Rotate_180} • kClip_Rotate_270 = { kClip_Rotate_270}

See Also

getRotateDegree()

Since

1.0.1

4.30.3.68 boolean com.nexstreaming.nexeditorsdk.nexClip.setSaturation (int *value*)

This method sets the saturation of a clip.

Example code:

```
project.getClip(1 + i, true).setSaturation(-25);
```

Parameters

value	The saturation to set on the clip, as an `integer`. This must be in the range of -255 to 255.

Returns

`TRUE` if the value is set successfully; `FALSE` if the value is not in range.

See Also

getSaturation()

Since

version 1.1.0

4.30.3.69 boolean com.nexstreaming.nexeditorsdk.nexClip.setSolidColor (int *color*)

This method changes the color of a clip only created by the method `getSolidClip()`.

Example code:

```
boolean isColor = clip.setSolidColor(Color.BLUE);
```

Parameters

color	The ARGB value of a new color to set to a clip as integer.

Returns

A clip with a new solid background color; `FALSE` if the clip was not created by `getSolidClip()`.

See Also

getSolidColor()

Since

version 1.3.43

4.30.3.70 static void com.nexstreaming.nexeditorsdk.nexClip.setThumbTempDir (String *path*) `[static]`

Deprecated For internal use only. Please do not use.

4.30.3.71 static void com.nexstreaming.nexeditorsdk.nexClip.setVideoClipDetailThumbnailsDiskLimit (long *size*) `[static]`

Deprecated For internal use only. Please do not use.

4.30.3.72 void com.nexstreaming.nexeditorsdk.nexClip.setVignetteEffect (boolean *on*)

Returns

Since

1.5.43

4.30.3.73 void com.nexstreaming.nexeditorsdk.nexClip.setVoiceChangerFactor (int *factor*)

This method sets the voice changer factor which is used to modulate the audio of a clip.

Parameters

factor	Audio modulate value as an `integer`, in range of 0 $\sim$ 4. Possible values for voice changer factor : • 0 : Default • 1 : Chipmunk • 2 : Robot • 3 : Deep • 4 : Modulation

Since

version 1.5.15

See Also

getVoiceChangerFactor()

4.30.4 Member Data Documentation

4.30.4.1 final int com.nexstreaming.nexeditorsdk.nexClip.AVC_Profile_Baseline = 66 `[static]`

This indicates the H.264 including Base Line profile.

Since

version 1.3.4

4.30.4.2 final int com.nexstreaming.nexeditorsdk.nexClip.AVC_Profile_Extended = 88 `[static]`

This indicates the H.264 including Extended profile.

Since

version 1.3.4

4.30.4.3 final int com.nexstreaming.nexeditorsdk.nexClip.AVC_Profile_High = 100 `[static]`

This indicates the H.264 including High profile.

Since

version 1.3.4

4.30.4.4 final int com.nexstreaming.nexeditorsdk.nexClip.AVC_Profile_High10 = 100 [static]

This indicates the H.264 including High 10 profile.

Since

version 1.3.4

4.30.4.5 final int com.nexstreaming.nexeditorsdk.nexClip.AVC_Profile_High422 = 122 [static]

This indicates the H.264 including High 422 profile.

Since

version 1.3.4

4.30.4.6 final int com.nexstreaming.nexeditorsdk.nexClip.AVC_Profile_High444 = 244 [static]

This indicates the H.264 including High 444 profile.

Since

version 1.3.4

4.30.4.7 final int com.nexstreaming.nexeditorsdk.nexClip.AVC_Profile_Main = 77 [static]

This indicates the H.264 including Main profile.

Since

version 1.3.4

4.30.4.8 final int com.nexstreaming.nexeditorsdk.nexClip.AVC_Profile_Unknown = 0 [static]

This indicates the H.264 including an unknown profile.

Since

version 1.3.4

4.30.4.9 final int com.nexstreaming.nexeditorsdk.nexClip.kClip_NotSupported = 12 [static]

This member indicates that a clip is unsupported by the NexEditor™ SDK.

See Also

getSupportedResult()

Since

version 1.0.0

4.30.4.10 final int com.nexstreaming.nexeditorsdk.nexClip.kClip_NotSupported_AudioCodec = 2 [static]

This member indicates that a clip includes an audio file unsupported by the NexEditor™ SDK.

See Also

getSupportedResult()

Since

version 1.0.0

4.30.4.11 final int com.nexstreaming.nexeditorsdk.nexClip.kClip_NotSupported_AudioProfile = 3 [static]

This member indicates that a clip includes an unsupported audio profile.

See Also

getSupportedResult()

Since

version 1.0.0

4.30.4.12 final int com.nexstreaming.nexeditorsdk.nexClip.kClip_NotSupported_Container = 4 [static]

This member indicates that a clip is in a format unsupported by the NexEditor™ SDK.

See Also

getSupportedResult()

Since

version 1.0.0

4.30.4.13 final int com.nexstreaming.nexeditorsdk.nexClip.kClip_NotSupported_DurationTooShort = 6 [static]

This member indicates that a video or audio clip is too short to be supported by the NexEditor™ SDK.

See Also

getSupportedResult()

Since

version 1.0.0

4.30.4.14 final int com.nexstreaming.nexeditorsdk.nexClip.kClip_NotSupported_ResolutionTooHigh = 5 [static]

This member indicates that a clip is unsupported because the resolution is too high.

See Also

getSupportedResult()

Since

version 1.0.0

4.30.4.15 final int com.nexstreaming.nexeditorsdk.nexClip.kClip_NotSupported_ResolutionTooLow = 7 [static]

This member indicates that a clip is unsupported because the resolution is too low.

See Also

getSupportedResult()

Since

version 1.0.0

4.30.4.16 final int com.nexstreaming.nexeditorsdk.nexClip.kClip_NotSupported_VideoCodec = 9 [static]

This member indicates that a clip includes an unsupported video file.

See Also

getSupportedResult()

Since

version 1.0.0

4.30.4.17 final int com.nexstreaming.nexeditorsdk.nexClip.kClip_NotSupported_VideoFPS = 10 [static]

This member indicates that a clip has an unsupported video FPS.

See Also

getSupportedResult()

Since

version 1.0.0

4.30.4.18 final int com.nexstreaming.nexeditorsdk.nexClip.kClip_NotSupported_VideoLevel = 11 [static]

This member indicates that a clip has an unsupported video level.

See Also

getSupportedResult()

Since

version 1.0.0

4.30.4.19 final int com.nexstreaming.nexeditorsdk.nexClip.kClip_NotSupported_VideoProfile = 8 [static]

This member indicates that a clip includes an unsupported video profile.

See Also

getSupportedResult()

Since

version 1.0.0

4.30.4.20 final int com.nexstreaming.nexeditorsdk.nexClip.kClip_Rotate_0 = 0 [static]

This member indicates no clip rotation.

See Also

setRotateDegree(int)
getRotateDegree()

Since

version 1.0.1

4.30.4.21 final int com.nexstreaming.nexeditorsdk.nexClip.kClip_Rotate_180 = 180 [static]

This member indicates a rotation of a clip by 180 degrees counterclockwise.

See Also

setRotateDegree(int)
getRotateDegree()

Since

version 1.0.1

4.30.4.22 final int com.nexstreaming.nexeditorsdk.nexClip.kClip_Rotate_270 = 270 [static]

This member indicates a rotation of a clip by 270 degrees counterclockwise.

See Also

setRotateDegree(int)
getRotateDegree()

Since

version 1.0.1

4.30.4.23 final int com.nexstreaming.nexeditorsdk.nexClip.kClip_Rotate_90 = 90 [static]

This member indicates a rotation of a clip by 90 degrees counterclockwise.

See Also

setRotateDegree(int)
getRotateDegree()

Since

version 1.0.1

4.30.4.24 final int com.nexstreaming.nexeditorsdk.nexClip.kCLIP_Supported = 0 [static]

This member indicates that a clip is supported by the NexEditor™ SDK.

See Also

getSupportedResult()

Since

version 1.0.0

4.30.4.25 final int com.nexstreaming.nexeditorsdk.nexClip.kClip_Supported_NeedFPSTranscoding = 14 [static]

This member indicates a clip that needs to be changed to a lower FPS.

See Also

getSupportedResult()

Since

version 1.0.0

4.30.4.26 final int com.nexstreaming.nexeditorsdk.nexClip.kClip_Supported_NeedResolutionTranscoding = 13 [static]

This member indicates that a clip needs to be changed to a lower resolution in order to be edited in the NexEditor™ SDK.

See Also

getSupportedResult()

Since

version 1.0.0

4.30.4.27 final int com.nexstreaming.nexeditorsdk.nexClip.kClip_Supported_Unknown = 1 [static]

This member indicates that it is not known whether or not a clip is supported by the NexEditor™ SDK.

See Also

getSupportedResult()

Since

version 1.0.0

4.30.4.28 final int com.nexstreaming.nexeditorsdk.nexClip.kCLIP_TYPE_AUDIO = 0x00000003 [static]

Audio files with file formats such as MP3 and AAC.

See Also

getClipType()

Since

version 1.0.0

See Also

getClipType()

4.30.4.29 final int com.nexstreaming.nexeditorsdk.nexClip.kCLIP_TYPE_IMAGE = 0x00000001 [static]

Image files of photos or pictures.

See Also

getClipType()

Since

version 1.0.0

4.30.4.30 final int com.nexstreaming.nexeditorsdk.nexClip.kCLIP_TYPE_NONE = 0x00000000 [static]

The status of the clip has not yet been clarified.

See Also

getClipType()

Since

version 1.0.0

4.30.4.31 final int com.nexstreaming.nexeditorsdk.nexClip.kCLIP_TYPE_VIDEO = 0x00000004 [static]

Video files with file formats such as MP4.

See Also

getClipType()

Since

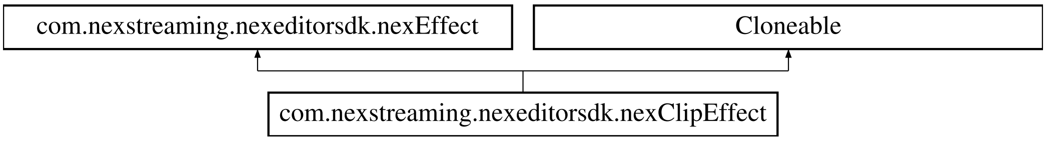
version 1.0.0

See Also

getClipType()

4.31 com.nexstreaming.nexeditorsdk.nexClipEffect Class Reference

Inheritance diagram for com.nexstreaming.nexeditorsdk.nexClipEffect:


com.nexstreaming.nexeditorsdk.nexEffect
Cloneable
com.nexstreaming.nexeditorsdk.nexClipEffect

Public Member Functions

- void setEffect (String ID)
- void setTitle (String str)
- String getTitle ()
- String getName (Context context)
- String getDesc (Context context)
- Bitmap getIcon ()
- void setShowStartTime (int time)
- void setShowEndTime (int time)
- void setEffectShowTime (int starttime, int endtime)
- int getShowStartTime ()
- int getShowEndTime ()
- void setAutoTheme ()
- String getCategoryTitle (Context context)

Static Protected Member Functions

- static nexClipEffect **clone** (nexClipEffect src)

Additional Inherited Members

4.31.1 Detailed Description

This class sets clip effects. A clip effect is an effect that can be set on a clip (to add dramatic visuals to the images or videos).

Different effects include different options that can be set and customized as desired.

Example code 1:

```
nexEffectLibrary fxlib = nexEffectLibrary.getEffectLibrary(this);
    nexClipEffect[] clipEffects = fxlib.getClipEffects();
    for (nexClipEffect e : clipEffects) {
        mEffectItem.add(e.getId());
    }
```

Example code 2:

class sample code

```
project.getClip(0,true).getClipEffect().setEffect(effectID);
```

See Also

nexEffectLibrary::getClipEffects()
nexEffectOptions
com.nexstreaming.nexeditorsdk.nexEffectLibrary::findClipEffectById(String)
com.nexstreaming.nexeditorsdk.nexClip::getClipEffect()

Since

version 1.0.0

4.31.2 Member Function Documentation

4.31.2.1 String com.nexstreaming.nexeditorsdk.nexClipEffect.getCategoryTitle (Context *context*)

This method gets the category name of a clip effect.

Example code:

```
name.setText(mClipEffects[index].getName(mContext)+":"+mClipEffects[index].
      getCategoryTitle(mContext));
```

Parameters

context	The Android context.

Returns

The category name as a `String`.

Since

version 1.3.3

4.31.2.2 String com.nexstreaming.nexeditorsdk.nexClipEffect.getDesc (Context *context*)

This method gets the description of an effect.

Example code:

```
name.setText(mClipEffects[index].getName(mContext)+":"+mClipEffects[index].
      getDesc(mContext));
```

Parameters

context	The Android context.

Returns

The description of an effect. If context is `null`, returns `null`.

See Also

getName(android.content.Context)

Since

version 1.0.0

4.31.2.3 Bitmap com.nexstreaming.nexeditorsdk.nexClipEffect.getIcon ( )

This method gets the icon of an effect.

Example code:

```
if( mClipEffects[index].getIcon() != null ){
                iv.setVisibility(View.VISIBLE);
                iv.setImageBitmap(mClipEffects[index].getIcon());
            }
```

Returns

An instance of the effect icon bitmap.

Since

version 1.0.0

4.31.2.4 String com.nexstreaming.nexeditorsdk.nexClipEffect.getName (Context *context*)

This method gets the name of an effect.

Example code:

```
name.setText(mClipEffects[index].getName(mContext)+":"+mClipEffects[index].
        getCategoryTitle(mContext));
```

Parameters

context	The Android context of the effect.

Returns

The name of the effect.

See Also

getDesc(android.content.Context)

Since

version 1.0.0

4.31.2.5 int com.nexstreaming.nexeditorsdk.nexClipEffect.getShowEndTime ( )

This method gets the end time of a clip effect.

Example code:

```
StringBuilder b = new StringBuilder();
        b.append(clip.getClipEffect().getShowStartTime());
        b.append(’,’);
        b.append(clip.getClipEffect().getShowEndTime());
        b.append(’?’);
```

Returns

The end time of a clip effect, in `msec` (milliseconds).

See Also

setShowEndTime(int)

Since

version 1.1.0

4.31.2.6 int com.nexstreaming.nexeditorsdk.nexClipEffect.getShowStartTime ( )

This method gets the start time of a clip effect.

Example code:

```
StringBuilder b = new StringBuilder();
b.append(clip.getClipEffect().getShowStartTime());
b.append(’,’);
b.append(clip.getClipEffect().getShowEndTime());
b.append(’?’);
```

Returns

The start time of a clip effect, in `msec` (milliseconds).

See Also

setShowStartTime(int)

Since

version 1.1.0

See Also

setShowStartTime(int)

4.31.2.7 String com.nexstreaming.nexeditorsdk.nexClipEffect.getTitle ( )

This method gets text from a clip effect.

Example code:

```
String title = project.getClip(0,true).getClipEffect().getTitle();
```

Returns

Text set to the clip effect.

See Also

setTitle(String)
setTitle(int, String)
getTitle(int)

Since

version 1.0.0

4.31.2.8 void com.nexstreaming.nexeditorsdk.nexClipEffect.setAutoTheme ( )

This method sets clip effects on a clip automatically, depending on the theme currently set.

Example code:

```
project.getClip(0,true).getClipEffect().setAutoTheme();
```

See Also

nexEffect::setEffectNone()
setEffect(String)

Since

version 1.0.0

4.31.2.9 void com.nexstreaming.nexeditorsdk.nexClipEffect.setEffect (String *ID*)

This method sets a clip effect, that is not part of any existing theme, on a clip. A clip effect that is not part of any theme can be brought from nexEffectLibrary#getClipEffects().

Example code:

```
mProject.getClip(i, true).getClipEffect().setEffect("com.nexstreaming.kmsdk.test.extendedpip");
```

Parameters

ID	The clip effect ID as a `String`.

Exceptions

com.nexstreaming.nexeditorsdk.exception.InvalidEffectIDException	Will occur if the ID is not a clip effect ID.

See Also

setAutoTheme()
setEffectNone()

Since

version 1.0.0

4.31.2.10 void com.nexstreaming.nexeditorsdk.nexClipEffect.setEffectShowTime (int *starttime,* int *endtime*)

This method sets the start time and end time of a clip effect. The clip effect's duration will automatically adjusted to the clip's duration if the effect duration set with this method is longer than the clip's duration.

Example code:

```
mProject.getClip(i, true).getClipEffect().setEffectShowTime(0,clip.getTotalTime()); //set full time effect
```

Parameters

starttime	The start time of the clip effect, in `msec` (milliseconds).
endtime	The end time of the clip effect, in `msec` (milliseconds).

Exceptions

InvalidRangeException	

Since

version 1.3.43

4.31.2.11 void com.nexstreaming.nexeditorsdk.nexClipEffect.setShowEndTime (int *time*)

This method sets the end time of a clip effect.

Example code:

```
project.getClip(0,true).getClipEffect().setShowEndTime(kClipDuration);
```

Parameters

time	The end time of the clip effect, in `msec` (milliseconds).

Exceptions

InvalidRangeException	

See Also

getShowEndTime()

Since

version 1.1.0

4.31.2.12 void com.nexstreaming.nexeditorsdk.nexClipEffect.setShowStartTime (int *time*)

This method sets the start time of a clip effect.

Example code:

```
project.getClip(0,true).getClipEffect().setShowStartTime(0);
```

Parameters

time	The start time of the clip effect, in `msec` (milliseconds).

Exceptions

InvalidRangeException	

See Also

getShowStartTime()

Since

version 1.1.0

4.31.2.13 void com.nexstreaming.nexeditorsdk.nexClipEffect.setTitle (String *str*)

This methods adds text to clip effects. If this is set to `null`, no text will appear on the screen

Example code:

```
project.getClip(0,true).getClipEffect().setTitle("Title test");
```

Parameters

str	The text to be printed on the screen, as a `String`.

See Also

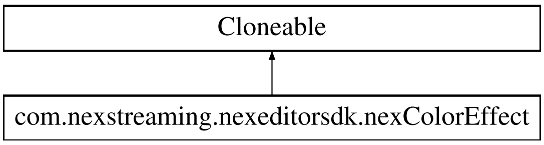
setTitle(int, String)
getTitle()
getTitle(int)

Since

version 1.0.0

4.32 com.nexstreaming.nexeditorsdk.nexColorEffect Class Reference

Inheritance diagram for com.nexstreaming.nexeditorsdk.nexColorEffect:



Public Member Functions

- nexColorEffect (float brightness, float contrast, float saturation, int tintColor)
- ColorMatrix getColorMatrix ()
- String getPresetName ()
- float getContrast ()
- float getBrightness ()
- float getSaturation ()
- int getTintColor ()
- int getLUTId ()
- boolean equals (Object o)
- int hashCode ()

Static Public Member Functions

- static List< nexColorEffect > getPresetList ()
- static nexColorEffect getLutColorEffect (String assetId)

  This gets the LUT Color Effect. When an Asset filter ID is entered, this gets the matching nexColorEffect.
- static void updatePluginLut ()

  An API for update after an LUT Asset is installed.
- static Bitmap applyColorEffectOnBitmap (Bitmap bgBitmap, nexColorEffect colorEffect)

Static Public Attributes

- static final nexColorEffect **NONE** = new nexColorEffect("NONE", 0, 0, 0, 0)
- static final nexColorEffect **ALIEN_INVASION** = new nexColorEffect("ALIEN_INVASION", 0.12f, -0.06f, -0.3f, 0xFF147014)
- static final nexColorEffect **BLACK_AND_WHITE** = new nexColorEffect("BLACK_AND_WHITE", 0f, 0f, -1f, 0)
- static final nexColorEffect **COOL** = new nexColorEffect("COOL", 0.12f, -0.12f, -0.3f, 0xFF144270)
- static final nexColorEffect **DEEP_BLUE** = new nexColorEffect("DEEP_BLUE", -0.2f, -0.3f, -0.6f, 0xFF0033FF)
- static final nexColorEffect **PINK** = new nexColorEffect("PINK", 0.1f, -0.3f, -0.6f, 0xFF9C4F4F)
- static final nexColorEffect **RED_ALERT** = new nexColorEffect("RED_ALERT", -0.30f, -0.19f, -1f, 0xFFF0000)
- static final nexColorEffect **SEPIA** = new nexColorEffect("SEPIA", 0.12f, -0.12f, -0.3f, 0xFF704214)
- static final nexColorEffect **SUNNY** = new nexColorEffect("SUNNY", 0.08f, -0.06f, -0.3f, 0xFFCCAA55)
- static final nexColorEffect **PURPLE** = new nexColorEffect("PURPLE", 0.08f, -0.06f, -0.3f, 0xFFAA55CC)
- static final nexColorEffect **ORANGE** = new nexColorEffect("ORANGE", 0.08f, -0.06f, -0.35f, 0xFFFFBB00)
- static final nexColorEffect **STRONG_ORANGE** = new nexColorEffect("STRONG_ORANGE", 0.08f, -0.06f, -0.5f, 0xFFFFBB00)
- static final nexColorEffect **SPRING** = new nexColorEffect("SPRING", 0.08f, -0.06f, -0.3f, 0xFFAACC55)
- static final nexColorEffect **SUMMER** = new nexColorEffect("SUMMER", 0.08f, -0.06f, -0.5f, 0xFFAAFF00)
- static final nexColorEffect **FALL** = new nexColorEffect("FALL", 0.08f, -0.06f, -0.5f, 0xFF00FFAA)
- static final nexColorEffect **ROUGE** = new nexColorEffect("ROUGE", 0.08f, -0.06f, -0.6f, 0xFFFF5555)
- static final nexColorEffect **PASTEL** = new nexColorEffect("PASTEL", 0.08f, -0.06f, -0.5f, 0xFF555555)
- static final nexColorEffect **NOIR** = new nexColorEffect("NOIR", -0.25f, 0.6f, -1.0f, 0xFF776655)

Static Protected Member Functions

- static nexColorEffect **clone** (nexColorEffect src)

4.32.1 Detailed Description

This class defines the color effect information possible for a clip such as brightness, saturation, contrast, and tint color.

By using preset values(NONE, ALIEN_INVASION, BLACK_AND_WHITE, COOL, DEEP_BLUE, PINK, RED_ALERT, SEPIA, SUNNY, PURPLE, ORANGE, STRONG_ORANGE, SPRING, SUMMER, FALL, ROUGE, PASTEL, NOIR), color effect settings can be used more efficiently.

For Example :

```
mEngine.getProject().getClip(0,true).setColorEffect( nexColorEffect.getPresetList().get(position) );
         mEngine.getProject().getClip(1,true).setColorEffect( nexColorEffect.COOL);
```

Since

version 1.0.1

4.32.2 Constructor & Destructor Documentation

4.32.2.1 com.nexstreaming.nexeditorsdk.nexColorEffect.nexColorEffect (float *brightness,* float *contrast,* float *saturation,* int *tintColor*)

This method creates a color effect based on the values of brightness, contrast, saturation and tint color passed with this method.

Example code:

```
public static final nexColorEffect ALIEN_INVASION = new nexColorEffect("ALIEN_INVASION", 0.12f, -0.06f, -0.
      3f, 0xFF147014);
```

Parameters

brightness	The brightness value as a `float(-1 ~ 1)`.
contrast	The contrast value as a `float(-1 ~ 1)`.
saturation	The saturation value as a `float(-1 ~ 1)`.
tintColor	The tint color value as an `integer(ARGB color format)`.

Since

version 1.1.0

4.32.3 Member Function Documentation

4.32.3.1 static Bitmap com.nexstreaming.nexeditorsdk.nexColorEffect.applyColorEffectOnBitmap (Bitmap *bgBitmap,* nexColorEffect *colorEffect*) `[static]`

This method is a preview of a color effect that uses the LUT filter.

Other color effects that are non-LUT can be previewd using the getColorMatrix().

Example code:

```
Bitmap thumbnail = Bitmap.createBitmap(...);
  nexColorEffect effect = nexColorEffect.getPresetList().get(20);
  int lut_id = effect.getLUTId();
  if(  lut_id != 0 ) {
      Bitmap bm = nexColorEffect.applyColorEffectOnBitmap(thumbnail, effect);
  }
```

Parameters

bgBitmap	The original image.
colorEffect	The Color effect with the LUT filter.

Returns

A LUT-filtered image; otherwise, the original image.

Since

version 1.5.23

4.32.3.2 boolean com.nexstreaming.nexeditorsdk.nexColorEffect.equals (Object *o*)

This method compares whether the color effect options set to two different methods are equal.

Parameters

o	`nexColorEffect`

Returns

`TRUE` if the setting options are equal, otherwise `FALSE`.

Since

version 1.1.0

4.32.3.3 float com.nexstreaming.nexeditorsdk.nexColorEffect.getBrightness ( )

This method gets the brightness of a clip.

Example code:

```
if( mColorEffect==null )
        return m_Brightness + (int)(255*mColorEffect.getBrightness());
```

Returns

The brightness value, as a `float`.

Since

version 1.1.0

4.32.3.4 ColorMatrix com.nexstreaming.nexeditorsdk.nexColorEffect.getColorMatrix ( )

This method gets the ColorMatrix of the Color effect that is not LUT.

Example code:

```
ImageView iv = (ImageView)convertView.findViewById(R.id.imageView_color_effect_apply);
iv.setColorFilter(new ColorMatrixColorFilter(nexColorEffect.COOL.getColorMatrix()));
```

Returns

android ColorMatrix

Since

version 1.5.23

4.32.3.5 float com.nexstreaming.nexeditorsdk.nexColorEffect.getContrast ( )

This method gets the contrast of a clip.

Example code:

```
if( mColorEffect==null )
        return m_Contrast + (int)(255*mColorEffect.getContrast());
```

Returns

The contrast value, as a `float`.

Since

version 1.1.0

4.32.3.6 static nexColorEffect com.nexstreaming.nexeditorsdk.nexColorEffect.getLutColorEffect (String *assetId*) [static]

This gets the LUT Color Effect. When an Asset filter ID is entered, this gets the matching nexColorEffect.

Parameters

assetId	Asset filter ID

Returns

nexColorEffect

Since

1.7.0

4.32.3.7 int com.nexstreaming.nexeditorsdk.nexColorEffect.getLUTId ( )

This method passes the LUT ID if the ColorEffect has LUT enabled.

Returns

The ID number if LUT is enabled; otherwise, 0.

Since

version 1.5.23

4.32.3.8 static List<nexColorEffect> com.nexstreaming.nexeditorsdk.nexColorEffect.getPresetList ( ) [static]

This method gets a list of the color effect presets already created.

Example code:

```
List<nexColorEffect> list = nexColorEffect.getPresetList();
        for( nexColorEffect ce: list ){
            items.add(ce.getPresetName());
        }
```

Returns

A list of the color presets, including:

- NONE, ALIEN_INVASION, BLACK_AND_WHITE, COOL, DEEP_BLUE, PINK,
- RED_ALERT, SEPIA, SUNNY, PURPLE, ORANGE, STRONG_ORANGE,
- SPRING, SUMMER, FALL, ROUGE, PASTEL, NOIR

See Also

getPresetName()

Since

version 1.0.1

4.32.3.9 String com.nexstreaming.nexeditorsdk.nexColorEffect.getPresetName ( )

This method gets the name of a preset color effect.

Example code:

```
for( nexColorEffect ce: list ){
            items.add(ce.getPresetName());
        }
```

Returns

The name of the color effect preset, as a `String`.

See Also

getPresetList()

Since

version 1.1.0

4.32.3.10 float com.nexstreaming.nexeditorsdk.nexColorEffect.getSaturation ( )

This method gets the saturation of a clip.

Example code:

```
if( mColorEffect==null )
        return m_Saturation + (int)(255*mColorEffect.getSaturation());
```

Returns

The saturation value, as a `float`.

Since

version 1.1.0

4.32.3.11 int com.nexstreaming.nexeditorsdk.nexColorEffect.getTintColor ( )

This method gets the tint color of a clip.

Example code:

```
if( mColorEffect==null )
        return  mColorEffect.getTintColor();
```

Returns

The tint color value, as an `integer`.

Since

version 1.1.0

4.32.3.12 int com.nexstreaming.nexeditorsdk.nexColorEffect.hashCode ( )

This method returns the unique value of a `nexColorEffect` object.

Returns

The unique value of a `nexColorEffect` object.

Since

version 1.1.0

4.32.3.13 static void com.nexstreaming.nexeditorsdk.nexColorEffect.updatePluginLut ( ) `[static]`

An API for update after an LUT Asset is installed.

Since

1.7.0

4.33 com.nexstreaming.nexeditorsdk.nexConfig Class Reference

Static Public Member Functions

- static void set (int HWCodecMemSize, int HWDecCount, int MaxFPS, boolean supportMPEGV4, int MaxResolution)
- static void **setCapability** (InputStream inputStream)
- static int getProperty (int key)
- static boolean setProperty (int key, int value)

Static Public Attributes

- static final int kSetUserConfig = 0
- static final int kHardwareCodecMemSize = 1
- static final int kHardwareDecMaxCount = 2
- static final int kMaxSupportedFPS = 3
- static final int kSupportMPEGV4 = 4
- static final int kMaxResolution = 5
- static final int **kForceDirectExport** = 6

4.33.1 Detailed Description

This class manages configuration to set additional settings for the NexEditor™ SDK.

This class should be called before init(android.content.Context, String) is called when starting NexEditor SDK.

Since

version 1.3.0

4.33.2 Member Function Documentation

4.33.2.1 static int com.nexstreaming.nexeditorsdk.nexConfig.getProperty (int *key*) [static]

This method gets the setting values of the NexEditor™ SDK.

Example code:

```
int value = nexConfig.getProperty(1);
```

Parameters

key	The value indicating each of the keys (1 ∼ 5).

Returns

The value set to the key.

See Also

setProperty(int, int)

Since

version 1.3.0

4.33.2.2 static void com.nexstreaming.nexeditorsdk.nexConfig.set (int *HWCodecMemSize,* int *HWDecCount,* int *MaxFPS,* boolean *supportMPEGV4,* int *MaxResolution*) [static]

This method sets the basic settings for the NexEditor™ SDK.

This should be called before init(android.content.Context, String) is called.

Example code:

```
nexConfig.set(3840 * 2160 * 3 / 2, 4, 120, false);
```

Parameters

HWCodecMemSize	The memory size of the hardware codec. The default value is 1920∗1088∗3/2.
HWDecCount	The number of decoders available at the same time. The default value is 2.
MaxFPS	The value of FPS. The default value is 120.
supportMPEGV4	Whether or not MPEG4 is supported. The default value is 0.
MaxResolution	The maximum resolution of the content to be added to the project. The Default setting is UHD.

See Also

setProperty(int, int)
getProperty(int)

Since

version 1.3.0

4.33.2.3 static boolean com.nexstreaming.nexeditorsdk.nexConfig.setProperty (int *key*, int *value*) [static]

This method changes the setting value of the NexEditor™ SDK.

This method should be called before init(android.content.Context, String) is called.

Example code:

```
nexConfig.setProperty(0,0);
```

Parameters

key	The value indicating each of the keys (1 ∼ 5).
value	The changed setting value of the key.

Returns

if key value is over property length return `FALSE`, else return `TRUE`.

See Also

getProperty(int)

Since

version 1.3.0

4.33.3 Member Data Documentation

4.33.3.1 final int com.nexstreaming.nexeditorsdk.nexConfig.kHardwareCodecMemSize = 1 [static]

This key sets the memory size of the hardware codec.

Values : Input memory size value as an `integer`. (ex. 3840∗2160∗3/2)

Default : 1920∗1088∗3/2

See Also

setProperty(int, int)
getProperty(int)

Since

version 1.3.0

4.33.3.2 final int com.nexstreaming.nexeditorsdk.nexConfig.kHardwareDecMaxCount = 2 [static]

This key sets the number of decoders, available at the same time, for the hardware codec.

Values : The number of decoders available at the same time. The maximum value is 2.

Default : 2

See Also

setProperty(int, int)
getProperty(int)

Since

version 1.3.0

4.33.3.3 final int com.nexstreaming.nexeditorsdk.nexConfig.kMaxResolution = 5 [static]

This sets the number of Max resolution of content.

Values: The number of Max resolution available as an `integer`.

Default : UHD

See Also

setProperty(int, int)
getProperty(int)

Since

version 1.5.15

4.33.3.4 final int com.nexstreaming.nexeditorsdk.nexConfig.kMaxSupportedFPS = 3 [static]

This sets the number of FPS available for decoding.

Values: The number of FPS available as an `integer`.

Default : 120

See Also

setProperty(int, int)
getProperty(int)

Since

version 1.3.0

4.33.3.5 final int com.nexstreaming.nexeditorsdk.nexConfig.kSetUserConfig = 0 [static]

This key decides whether the NexEditor™ SDK should operate with the `nexConfig` settings or default settings of the SDK.

Values :

- 0 - Operates NexEditor™ SDK with default settings.
- 1 - Operates NexEditor™ SDK with settings set with `nexConfig.setProperty`.

See Also

setProperty(int, int)
getProperty(int)

Since

version 1.3.0

4.33.3.6 final int com.nexstreaming.nexeditorsdk.nexConfig.kSupportMPEGV4 = 4 [static]

This decides whether the NexEditor™ SDK should support MPEG4 if the hardware codec supports MPEG4.

values :

- 0 - Do not support MPEG4.
- 1 - Support MPEG4.

Default : 0

See Also

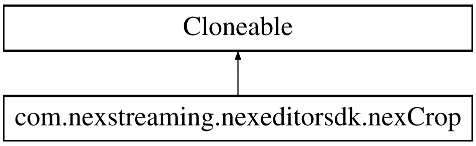
setProperty(int, int)
getProperty(int)

Since

version 1.3.0

4.34 com.nexstreaming.nexeditorsdk.nexCrop Class Reference

Inheritance diagram for com.nexstreaming.nexeditorsdk.nexCrop:



Classes

- enum CropMode

Public Member Functions

- int getRotate ()
- int getWidth ()
- int getHeight ()
- void randomizeStartEndPosition (boolean faceDetection, CropMode cropMode)
- void getStartPosition (Rect rect)
- void setStartPosition (Rect rect)
- void getEndPosition (Rect rect)
- void setEndPosition (Rect rect)
- void getStartPositionRaw (Rect rect)
- void getEndPositionRaw (Rect rect)
- void resetStartEndPosition ()
- void getFaceBounds (Rect bounds)

Static Public Attributes

- static final int ABSTRACT_DIMENSION = 100000

Static Protected Member Functions

- static nexCrop **clone** (nexCrop src)

4.34.1 Detailed Description

This class defines how the NexEditor™ SDK crops and actually displays video and image clips in a project.

When the start and end positions of a "crop" have been designated on a clip, the relevant clip will play from the start position of the crop to the end position of the crop. For images in a project, this creates a Ken Burns effect.

The start and end of a crop are defined in terms of the original size of the clip. The resolution of the cropped clip will be increased or decreased appropriately when the project is previewed or exported.

Example code :

int wid = mEngine.getProject().getClip(0, true).getCrop().getWidth(); int hei = mEngine.getProject().getClip(0, true).getCrop().getHeight(); int rot = mEngine.getProject().getClip(0, true).getCrop().getRotate();

if(rot == 90 || rot == 270) { mRatio = 360f / (float) wid; int tmp = wid; wid = hei; hei = tmp; }else{ mRatio = 360f / (float) hei; }

Rect startRect = new Rect(); mEngine.getProject().getClip(0, true).getCrop().getStartPosition(startRect);

startRect.left = (int) (startRect.left ∗ mRatio); startRect.top = (int) (startRect.top ∗ mRatio); startRect.right = (int) (startRect.right ∗ mRatio); startRect.bottom = (int) (startRect.bottom ∗ mRatio);

Rect endRect = new Rect(); mEngine.getProject().getClip(0, true).getCrop().getEndPosition(endRect);

endRect.left = (int) (endRect.left ∗ mRatio); endRect.top = (int) (endRect.top ∗ mRatio); endRect.right = (int) (endRect.right ∗ mRatio); endRect.bottom = (int) (endRect.bottom ∗ mRatio);

Since

version 1.1.0

4.34.2 Member Function Documentation

4.34.2.1 void com.nexstreaming.nexeditorsdk.nexCrop.getEndPosition (Rect *rect*)

This method gets the end crop position to be displayed on a clip, as a rectangle, `rect`.

Example code:

```
mEngine.getProject().getClip(0, true).getCrop().getEndPosition(endRect);
```

Parameters

rect	The end position for the crop on a clip, as a rectangle.

See Also

setStartPosition(android.graphics.Rect)
setEndPosition(android.graphics.Rect)
getStartPosition(android.graphics.Rect)

Since

version 1.1.0

4.34.2.2 void com.nexstreaming.nexeditorsdk.nexCrop.getEndPositionRaw (Rect *rect*)

This method gets the end position of a crop on a clip (as a rectangle), converted into a range value that's internally used by the NexEditor™ SDK. use to com.nexstreaming.nexeditorsdk.nexEngine#fastPreviewCrop(android.graphics.Rect)

Example code:

```
Rect raw = new Rect();
        mEngine.getProject().getClip(0, true).getCrop().getEndPositionRaw(raw);
```

Parameters

rect	The position of the rectangle, converted into the NexEditor™ SDK range value (0 ∼ 100000).

See Also

getStartPositionRaw(android.graphics.Rect)
com.nexstreaming.nexeditorsdk.nexEngine::fastPreviewCrop(android.graphics.Rect)

Since

version 1.3.43

4.34.2.3 void com.nexstreaming.nexeditorsdk.nexCrop.getFaceBounds (Rect *bounds*)

This method detects any face from the image item.

Example code:

```
mEngine.getProject().getClip(0, true).getCrop().getFaceBounds(raw);
```

Parameters

bounds	Detected face area on the image item, as `Rect`.

Since

version 1.3.27

4.34.2.4 int com.nexstreaming.nexeditorsdk.nexCrop.getHeight ( )

This method gets the height of the original clip, before a crop.

Example code:

```
int hei = mEngine.getProject().getClip(0, true).getCrop().getHeight();
```

Returns

The original height of the clip, as an `integer`.

See Also

getWidth()
nexClip::getHeight()

Since

version 1.1.0

4.34.2.5 int com.nexstreaming.nexeditorsdk.nexCrop.getRotate ( )

This method gets the degrees of rotation on a `nexClip` instance of a rotated clip where a crop will be performed.

This reflects the same value that is set by `setRotateDegree(int)` in `nexClip`.

Example code:

```
int rot = mEngine.getProject().getClip(0, true).getCrop().getRotate();
```

Returns

The degrees of rotation of the clip, where 0 is no rotation and for example 180 would mean the clip is upside down.

See Also

nexClip::getRotateDegree()

Since

version 1.1.0

4.34.2.6 void com.nexstreaming.nexeditorsdk.nexCrop.getStartPosition (Rect *rect*)

This method gets the starting crop position to be displayed on a clip, as a rectangle, `Rect`.

Example code:

```
mEngine.getProject().getClip(0, true).getCrop().getStartPosition(startRect);
```

Parameters

rect	The starting crop position to be shown, as a rectangle.

See Also

setStartPosition(android.graphics.Rect)
setEndPosition(android.graphics.Rect)
getEndPosition(android.graphics.Rect)

Since

version 1.1.0

4.34.2.7 void com.nexstreaming.nexeditorsdk.nexCrop.getStartPositionRaw (Rect *rect*)

This method gets the start position of a crop on a clip (as a rectangle), converted into a range value that's internally used by the NexEditor™ SDK. use to com.nexstreaming.nexeditorsdk.nexEngine#fastPreviewCrop(android.-graphics.Rect)

Example code:

```
Rect raw = new Rect();
        mEngine.getProject().getClip(0, true).getCrop().getStartPositionRaw(raw);
```

Parameters

rect	The position of the rectangle, converted into the NexEditor™ SDK range value (0 ∼ 100000).

See Also

getEndPositionRaw(android.graphics.Rect)
com.nexstreaming.nexeditorsdk.nexEngine::fastPreviewCrop(android.graphics.Rect)

Since

version 1.3.43

4.34.2.8 int com.nexstreaming.nexeditorsdk.nexCrop.getWidth ( )

This method gets the width of the original clip, before a crop.

Example code:

```
int wid = mEngine.getProject().getClip(0, true).getCrop().getWidth();
```

Returns

The original width of the clip, as an `integer`.

See Also

getHeight()
nexClip::getWidth()

Since

version 1.1.0

4.34.2.9 void com.nexstreaming.nexeditorsdk.nexCrop.randomizeStartEndPosition (boolean *faceDetection*, CropMode *cropMode*)

This method randomizes the start and end crops of a clip, based on the chosen crop mode.

By default, the NexEditor™ SDK randomly selects the start and end positions of crops on clips added, as if the `cropMode` parameter is set to `PAN_RAND`.

Parameters

faceDetection	Crop to sections where faces are detected in the clip (Not currently available but planned for future versions).
cropMode	The crop mode to use when randomly setting the start and end of a crop. This will be one of the following CropMode options: cropMode • **FIT:** The largest dimension of the original clip will be fit to the project display, without any distortion of the original dimensions (and the rest of the display will display black). • **FILL:** The clip will be cropped to fill the project display, and the original may be distorted. • **PAN_RAND:** The start and end positions of the crop will be randomly selected. • **PAN_FACE:** Not currently available.

Example code :

```
mEngine.getProject().getClip(0, true).getCrop().randomizeStartEndPosition(false,
       nexCrop.CropMode.FIT)
```

See Also

com.nexstreaming.nexeditorsdk.nexCrop.CropMode

Since

version 1.1.0

4.34.2.10 void com.nexstreaming.nexeditorsdk.nexCrop.resetStartEndPosition ( )

Initializes the start and end positions of the crop. Switches to Full Screen mode.

Since

1.5.42

4.34.2.11 void com.nexstreaming.nexeditorsdk.nexCrop.setEndPosition (Rect *rect*)

This method sets the end position of a crop on a clip.

The original size of the clip is used as the frame of reference for the crop.

Example code:

```
rect.left = (int) (rect.left / mRatio);
        rect.top = (int) (rect.top / mRatio);
        rect.right = (int) (rect.right / mRatio);
        rect.bottom = (int) (rect.bottom / mRatio);
        mEngine.getProject().getClip(0, true).getCrop().setStartPosition(rect);
```

Note

If, for example, the cropped section of the original should be displayed at a ratio of 16:9, then the crop coordinates should also be set to a ratio of 16:9.

Parameters

rect	The end position to set for the crop on a clip, as a rectangle.

See Also

setStartPosition(android.graphics.Rect)
getStartPosition(android.graphics.Rect)
getEndPosition(android.graphics.Rect)

Since

version 1.1.0

4.34.2.12 void com.nexstreaming.nexeditorsdk.nexCrop.setStartPosition (Rect *rect*)

This method sets the starting position of a crop on a clip.

The original size of the clip is used as the frame of reference for the crop.

Note

If, for example, the cropped section of the original should be displayed at a ratio of 16:9, then the crop coordinates should also be set to a ratio of 16:9.

Example code:

```
rect.left = (int) (rect.left / mRatio);
rect.top = (int) (rect.top / mRatio);
rect.right = (int) (rect.right / mRatio);
rect.bottom = (int) (rect.bottom / mRatio);
mEngine.getProject().getClip(0, true).getCrop().setStartPosition(rect);
```

Parameters

rect	The start position to set for the crop on a clip, as a rectangle.

See Also

setEndPosition(android.graphics.Rect)
getStartPosition(android.graphics.Rect)
getEndPosition(android.graphics.Rect)

Since

version 1.1.0

4.34.3 Member Data Documentation

4.34.3.1 final int com.nexstreaming.nexeditorsdk.nexCrop.ABSTRACT_DIMENSION = 100000 [static]

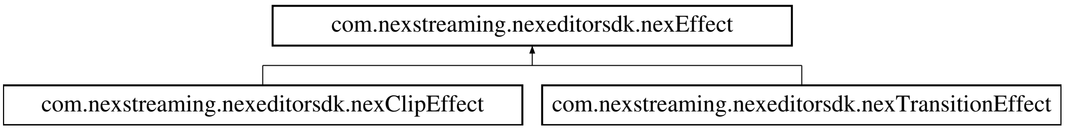
The value of the "Abstract" dimension used when cropping.

Since

version 1.1.0

4.35 com.nexstreaming.nexeditorsdk.nexEffect Class Reference

Inheritance diagram for com.nexstreaming.nexeditorsdk.nexEffect:


com.nexstreaming.nexeditorsdk.nexEffect
com.nexstreaming.nexeditorsdk.nexClipEffect
com.nexstreaming.nexeditorsdk.nexTransitionEffect

Public Member Functions

- int getType ()
- String getId ()
- int getDuration ()
- void setDuration (int duration)
- void setEffectNone ()
- void setTitle (int TrackNum, String str)
- String getTitle (int TrackNum)
- boolean updateEffectOptions (nexEffectOptions options, boolean up2Effect)

Static Public Member Functions

- static String getTitleOptions (nexEffectOptions options)

Static Public Attributes

- static final int kEFFECT_NONE = 0
- static final int kEFFECT_CLIP_AUTO = 1
- static final int kEFFECT_CLIP_USER = 2
- static final int kEFFECT_TRANSITION_AUTO = 3
- static final int kEFFECT_TRANSITION_USER = 4
- static final int **kEFFECT_OVERLAY_FILTER** = 5

4.35.1 Detailed Description

This class defines a NexEditor™ effect that can be applied to clips in a project.

Since

version 1.0.0

4.35.2 Member Function Documentation

4.35.2.1 int com.nexstreaming.nexeditorsdk.nexEffect.getDuration ( )

This method gets the duration of an effect.

Returns

The duration of the effect as an `integer` in `msec` (milliseconds).

Since

version 1.0.0

4.35.2.2 String com.nexstreaming.nexeditorsdk.nexEffect.getId ( )

This method gets the ID of an effect.

Returns

`none` if no effect was set, otherwise the ID of the effect as a `String`.

Since

version 1.0.0

4.35.2.3 String com.nexstreaming.nexeditorsdk.nexEffect.getTitle (int *TrackNum*)

This method gets the input text.

Parameters

TrackNum	The index of text tracks(maximum of 4 tracks) from 0 ∼ 3, within a field.

Returns

The text from the current track.

Since

version 1.0.0

See Also

setTitle(int, String)

4.35.2.4 static String com.nexstreaming.nexeditorsdk.nexEffect.getTitleOptions (nexEffectOptions *options*) [static]

This method gets the title effect options as `Strings`, from the serialized effect option values.

Parameters

options	The value of the serialized effect options.

Returns

The value of the effect options as `Strings`.

Example code :

```
m_previewView.setEffectOptions(nexEffect.getTitleOptions(mOptions));
m_previewView.setEffect(mSelectedEffectId );
m_previewView.setEffectTime(3000);
```

See Also

com.nexstreaming.nexeditorsdk.nexEffectPreviewView::setEffectOptions(String)
com.nexstreaming.nexeditorsdk.nexEffectLibrary::getEffectOptions(android.content.Context, String)

Since

version 1.1.0

4.35.2.5 int com.nexstreaming.nexeditorsdk.nexEffect.getType ( )

This method gets the type of an effect.

Returns

The type of clip effect. This will be one of:

- kEFFECT_NONE = { kEFFECT_NONE}
- kEFFECT_CLIP_AUTO = { kEFFECT_CLIP_AUTO}
- kEFFECT_CLIP_USER = { kEFFECT_CLIP_USER}
- kEFFECT_TRANSITION_AUTO = { kEFFECT_TRANSITION_AUTO}
- kEFFECT_TRANSITION_USER = { kEFFECT_TRANSITION_USER}

Since

version 1.0.0

4.35.2.6 void com.nexstreaming.nexeditorsdk.nexEffect.setDuration (int *duration*)

This method sets the duration of an effect.

Parameters

duration	The duration of the effect as an `integer` in `msec` (milliseconds).

Since

version 1.0.0

4.35.2.7 void com.nexstreaming.nexeditorsdk.nexEffect.setEffectNone ( )

This method removes an effect that has been set on a clip.

Since

version 1.0.0

4.35.2.8 void com.nexstreaming.nexeditorsdk.nexEffect.setTitle (int *TrackNum,* String *str*)

This method sets the input text for effects with text. Depending on the effects, the number of text tracks may vary.

Parameters

TrackNum	The index of text tracks(maximum of 4 tracks) from 0 ∼ 3, within a field.
str	The input text.

Since

version 1.0.0

See Also

getTitle(int)

4.35.2.9 boolean com.nexstreaming.nexeditorsdk.nexEffect.updateEffectOptions (nexEffectOptions *options,* boolean *up2Effect*)

This method updates the effect options in a project.

Parameters

options	The effect option to update to the project.
up2Effect	`TRUE` to get updated effect option, or set to `FALSE` to get the default effect option.

Returns

`TRUE` if the ID of the current effect and the effect that the option is updated from are the same, otherwise `FALSE`.

Example code :

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
     super.onActivityResult(requestCode, resultCode, data);
     if(requestCode == 100 && resultCode == Activity.RESULT_OK) {
         nexEffectOptions opt = (nexEffectOptions)data.getSerializableExtra("effectopt");
         String effect = data.getStringExtra("effect");
         mEngine.getProject().getClip(0,true).getClipEffect().setEffect(effect);
         mEngine.getProject().getClip(0,true).getClipEffect().
      updateEffectOptions(opt, true);
         mEngine.updateProject();
     }
 }
```

See Also

com.nexstreaming.nexeditorsdk.nexEffectLibrary::getEffectOptions(android.content.Context, String)

Since

version 1.0.1

4.35.3 Member Data Documentation

4.35.3.1 final int com.nexstreaming.nexeditorsdk.nexEffect.kEFFECT_CLIP_AUTO = 1 [static]

This member indicates that the clip effect is automatically set matching the theme currently set.

See Also

getType()

Since

version 1.0.0

4.35.3.2 final int com.nexstreaming.nexeditorsdk.nexEffect.kEFFECT_CLIP_USER = 2 [static]

This member indicates that the clip effect is not part of any theme but can be set by the user.

See Also

getType()

Since

version 1.0.0

4.35.3.3 final int com.nexstreaming.nexeditorsdk.nexEffect.kEFFECT_NONE = 0 [static]

This member indicates that no effect is set to a clip.

See Also

getType()

Since

version 1.0.0

4.35.3.4 final int com.nexstreaming.nexeditorsdk.nexEffect.kEFFECT_TRANSITION_AUTO = 3 [static]

This member indicates that the transition effect is automatically set matching the theme currently set.

See Also

getType()

Since

version 1.0.0

4.35.3.5 final int com.nexstreaming.nexeditorsdk.nexEffect.kEFFECT_TRANSITION_USER = 4 [static]

This member indicates that the transition effect is set by the user.

See Also

getType()

Since

version 1.0.0

4.36 com.nexstreaming.nexeditorsdk.nexEffectLibrary Class Reference

Classes

- class OnInstallPluginEffectPackageAsyncListener

Public Member Functions

- ArrayList< nexTheme > getThemesEx ()
- nexTheme[] getThemes ()
- ArrayList< nexTransitionEffect > getTransitionEffectsEx ()
- nexTransitionEffect[] getTransitionEffects ()
- nexClipEffect[] getClipEffects ()
- nexOverlayFilter[] getOverlayFilters ()

  This method gets Overlay filters. The Overlay filter is inserted in the Overlay track of a project. In other words, it is an Overlay transition.
- nexTheme findThemeById (String id)
- nexClipEffect findClipEffectById (String id)
- nexTransitionEffect findTransitionEffectById (String id)
- nexOverlayFilter findOverlayFilterById (String id)

  If the known ID is an overlay filter, this method creates and returns an instance of nexOverlayFiler.
- nexEffectOptions getEffectOptions (Context context, String id)
- boolean **installPluginEffectPackageAsync** (String[] pluginKmtFileNames, final OnInstallPluginEffectPackageAsyncListener listener)

Static Public Member Functions

- static nexEffectLibrary getEffectLibrary (Context context)
- static String **getPluginDirPath** ()

4.36.1 Detailed Description

This class sets clip effects on clips. To run this class properly, call init}(android.content.Context, String) first. This operates using a singleton, therefore there is no need to create an instance. Refer to the sample code below.

Example code :

nexEffectLibrary fxLib = nexEffectLibrary.getEffectLibrary(this);

Since

version 1.0.0

See Also

waitForLoading}(android.content.Context, Runnable)

4.36.2 Member Function Documentation

4.36.2.1 nexClipEffect com.nexstreaming.nexeditorsdk.nexEffectLibrary.findClipEffectById (String *id*)

This method gets a clip effect using a clip effect ID if the ID is known.

Example code:

```
Effect effect = EditorGlobal.getEditor().getEffectLibrary().findEffectById(
      options.getEffectID());
```

Parameters

id	The clip effect ID as a `String`.

Returns

`nexClipEffect`; `null` if no clip effect was found.

Since

version 1.0.0

See Also

findTransitionEffectById(String)

4.36.2.2 nexOverlayFilter com.nexstreaming.nexeditorsdk.nexEffectLibrary.findOverlayFilterById (String *id*)

If the known ID is an overlay filter, this method creates and returns an instance of nexOverlayFiler.

Parameters

id	overlay filter ID.

Returns

null – ID does not exist or it is not an Overlayfilter ID.

Since

1.7.0

4.36.2.3 nexTheme com.nexstreaming.nexeditorsdk.nexEffectLibrary.findThemeById (String *id*)

This method gets a theme package using a theme ID if the ID is known.

Example code:

```
theme = fxLib.findThemeById(getThemeId());
```

Parameters

id	The theme ID as a `String`.

Returns

`nexTheme`; `null` if no theme was found.

Since

version 1.0.0

4.36.2.4 nexTransitionEffect com.nexstreaming.nexeditorsdk.nexEffectLibrary.findTransitionEffectById (String *id*)

This method get a transition effect with a transition effect ID if the ID is known.

Example code:

```
Effect transitionEffect = findTransitionEffectById( id );
```

Parameters

id	The transition effect ID as a `String`.

Returns

`nexTransitionEffect`

Since

version 1.0.0

See Also

findClipEffectById(String)

4.36.2.5 nexClipEffect [] com.nexstreaming.nexeditorsdk.nexEffectLibrary.getClipEffects ( )

This method gets a basic clip effects that are not part of a theme package.

Example code :

```
nexEffectLibrary fxlib = nexEffectLibrary.getEffectLibrary(this);
    nexClipEffect[] clipEffects = fxlib.getClipEffects();
    for (nexClipEffect e : clipEffects) {
        Log.i(LOG_TAG,"Name =" + e.getName(getBaseContext()));
    }
```

Returns

An array of `nexClipEffect`.

Since

version 1.0.0

See Also

getThemes()
getTransitionEffects()

4.36.2.6 static nexEffectLibrary com.nexstreaming.nexeditorsdk.nexEffectLibrary.getEffectLibrary (Context *context*) [static]

This method gets the library handle for clip effects, transition effects, and theme packages needed for the editing.

Example code:

```
private nexEffectLibrary mfxLib;
        mfxLib = nexEffectLibrary.getEffectLibrary(context);
```

Parameters

context	Context of the activity.

Returns

`nexEffectLibrary`

Since

version 1.0.0

4.36.2.7 nexEffectOptions com.nexstreaming.nexeditorsdk.nexEffectLibrary.getEffectOptions (Context *context*, String *id*)

This method gets the effect options to set to clip effects made by the developer.

Parameters

context	The Android context.
id	An ID of an effect.

Returns

The instance of nexEffectOptions.

Example code :

```
 nexEffectLibrary fxlib = nexEffectLibrary.getEffectLibrary(this);
nexEffectOptions options = fxlib.getEffectOptions(this, transitionid);
if( options != null ) {
    int selectCount = 0;
    int lastIndex = 0;
    if (options.getSelectOptions().size() > 0) {
        selectCount = options.getSelectOptions().get(0).getItems().length;
    }
    for (int i = 0; i < project.getTotalClipCount(true); i++) {
        project.getClip(i, true).getTransitionEffect().setTransitionEffect(transitionid);
        if (selectCount > 0) {
            options.getSelectOptions().get(0).setSelectIndex(lastIndex);
            lastIndex++;
            if (lastIndex >= selectCount) lastIndex = 0;

            project.getClip(i, true).getTransitionEffect().updateEffectOptions(options, true);
        }
    }
}
```

See Also

updateEffectOptions(nexEffectOptions, boolean)
getTitleOptions(nexEffectOptions)

Since

version 1.0.1

4.36.2.8 nexOverlayFilter [ ] com.nexstreaming.nexeditorsdk.nexEffectLibrary.getOverlayFilters ( )

This method gets Overlay filters. The Overlay filter is inserted in the Overlay track of a project. In other words, it is an Overlay transition.

Returns

Overlay filter

Since

1.7.0

4.36.2.9 nexTheme [ ] com.nexstreaming.nexeditorsdk.nexEffectLibrary.getThemes ( )

This method gets registered theme packages in sequence.

Example code :

```
nexEffectLibrary fxlib = nexEffectLibrary.getEffectLibrary(this);
    nexTheme[] allThemes = fxlib.getThemes();
    for( nexTheme t: allThemes ) {
         Log.i(LOG_TAG,"Name =" + t.getName(getBaseContext()));
    }
```

Returns

The sequence of `nexTheme`.

Since

version 1.0.0

See Also

getTransitionEffects()
getClipEffects()

4.36.2.10 ArrayList<nexTheme> com.nexstreaming.nexeditorsdk.nexEffectLibrary.getThemesEx ( )

This method gets the registered theme packages as an ArrayList.

Example code:

```
mThemeTab = mfxLib.getThemesEx();
```

Returns

The ArrayList of `nexTheme`.

Since

version 1.0.0

See Also

getTransitionEffects()
getClipEffects()

4.36.2.11 nexTransitionEffect [] com.nexstreaming.nexeditorsdk.nexEffectLibrary.getTransitionEffects ( )

This method gets a basic transition effects that are not part of a theme package.

Example code :

```
nexEffectLibrary fxlib = nexEffectLibrary.getEffectLibrary(this);
    nexTransitionEffect[] transitionEffects = fxlib.getTransitionEffects();
    for (nexTransitionEffect e : transitionEffects) {
        Log.i(LOG_TAG,"Name =" + e.getName(getBaseContext()));
    }
```

Returns

An array of `nexTransitionEffect`.

Since

version 1.0.0

See Also

getThemes()
getClipEffects()

4.36.2.12 ArrayList<nexTransitionEffect> com.nexstreaming.nexeditorsdk.nexEffectLibrary.getTransitionEffectsEx ( )

This method gets the basic transition effects that are not part of a theme package.

Example code:

```
mTransitionTab = mfxLib.getTransitionEffectsEx();
```

Returns

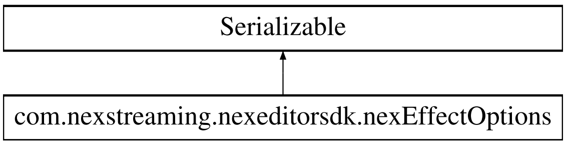
An array of `nexTransitionEffect` effects.

Since

version 1.0.0

4.37 com.nexstreaming.nexeditorsdk.nexEffectOptions Class Reference

Inheritance diagram for com.nexstreaming.nexeditorsdk.nexEffectOptions:



Classes

- class ColorOpt
- class Option
- class RangeOpt
- class SelectOpt
- class SwitchOpt
- class TextOpt

Public Member Functions

- String getEffectID ()
- List< TextOpt > getTextOptions ()
- List< ColorOpt > getColorOptions ()
- List< SelectOpt > getSelectOptions ()
- List< RangeOpt > getRangeOptions ()
- List< SwitchOpt > getSwitchOptions ()
- void setDefaultValue ()
- int getTextFieldCount ()

Protected Member Functions

- **nexEffectOptions** (String effectID, int type)
- void **addTextOpt** (String id, String label, int lineNum)
- void **addColorOpt** (String id, String label, String color)
- void **addSelectOpt** (String id, String label, String[ ] items, String[ ] values, int select)
- void **addRangeOpt** (String id, String label, int default_value, int min_value, int max_value)
- void **addSwitchOpt** (String id, String label, boolean on)

4.37.1 Detailed Description

This class defines and sets the options available for an effect on a clip.

Each effect in the NexEditor™ SDK has different characteristics and therefore different possible settings options with default values, but the default settings can be adjusted in detail using this class to customize the effects.

Instances of this class cannot be created as stand-alone instances but must be used with getEffectOptions(android.content.Context, String).

For Example:

```
nexEffectOptions opt = (nexEffectOptions)data.getSerializableExtra("effectopt");
        String effect = data.getStringExtra("effect");
        mEngine.getProject().getClip(0,true).getClipEffect().setEffect(effect);
        mEngine.getProject().getClip(0,true).getClipEffect().updateEffectOptions(opt, true);
        mEngine.updateProject();
```

Since

version 1.0.1

4.37.2 Member Function Documentation

4.37.2.1 List<ColorOpt> com.nexstreaming.nexeditorsdk.nexEffectOptions.getColorOptions ( )

This method retrieves the color-related settings and options available on an effect, as a list.

Possible color-related options for an effect include text color, background color, or effect color but the options available depend on the specific effect.

Example code:

```
mOptions.getColorOptions().get(0).setARGBColor(mColor1Value);
```

Returns

A list of the color-related options that can be set on the effect.

See Also

getTextOptions()
getSelectOptions()

Since

version 1.0.1

4.37.2.2 String com.nexstreaming.nexeditorsdk.nexEffectOptions.getEffectID ( )

This method gets the ID of an effect, as a `String`.

Example code:

```
Effect effect = EditorGlobal.getEditor().getEffectLibrary().findEffectById(options.getEffectID());
```

Returns

The effect ID as a `String`.

Since

version 1.1.0

4.37.2.3 List<**RangeOpt**> com.nexstreaming.nexeditorsdk.nexEffectOptions.getRangeOptions ( )

This gets the RangeOpt list from the options.

Returns

Since

1.7.0

4.37.2.4 List<SelectOpt> com.nexstreaming.nexeditorsdk.nexEffectOptions.getSelectOptions ( )

This method gets the designated text location and other select options available on an effect.

Example code:

```
mOptions.getSelectOptions().get(0).setSelectIndex(position);
```

Returns

The list of select options that can be set on the effect.

See Also

getTextOptions()
getColorOptions()

Since

version 1.0.1

4.37.2.5 List<SwitchOpt> com.nexstreaming.nexeditorsdk.nexEffectOptions.getSwitchOptions ( )

This gets the SwitchOpt list from the options.

Returns

Since

1.7.0

4.37.2.6 int com.nexstreaming.nexeditorsdk.nexEffectOptions.getTextFieldCount ( )

This method gets the number of text fields in an effect where text can be entered.

Example code:

```
for( int line = 0; line < mOptions.getTextFieldCount() ; line++ ){}
```

Returns

The number of text fields in the effect, as an `integer`.

Since

version 1.0.1

4.37.2.7 List<TextOpt> com.nexstreaming.nexeditorsdk.nexEffectOptions.getTextOptions ( )

This method gets the input text and other text-related options for an effect.

Example code:

```
mOptions.getTextOptions().get(0).setText(mLine1.getText().toString());
```

Returns

All text options related to an effect, as a list.

See Also

getColorOptions()
getSelectOptions()

Since

version 1.1.0

4.37.2.8 void com.nexstreaming.nexeditorsdk.nexEffectOptions.setDefaultValue ( )

This method resets the color and select options that can be set on an effect back to their original default values.

Example code:

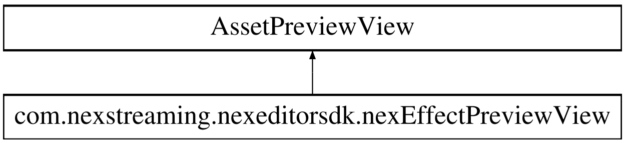
```
mOptions.setDefaultValue();
```

Since

version 1.1.0

4.38 com.nexstreaming.nexeditorsdk.nexEffectPreviewView Class Reference

Inheritance diagram for com.nexstreaming.nexeditorsdk.nexEffectPreviewView:



Public Member Functions

- **nexEffectPreviewView** (Context context, AttributeSet attrs)
- void setEffect (String effect_id)
- void setEffectTime (int effectTime)
- void setEffectOptions (String effectOptions)

4.38.1 Detailed Description

This class defines and enables the preview of an effect before it is actually applied to a clip. **Layout XML**

```
<com.nexstreaming.nexeditorsdk.nexEffectPreviewView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/effectPreview"
    android:layout_gravity="center_horizontal" />
```

class sample code

```
nexEffectPreviewView previewView = (nexEffectPreviewView)findViewById(R.id.XXXXX);
       previewView.setEffectTime(3000);
       String[] aStrings = new String[2];
       aStrings[0] = "1st Text Track";
       aStrings[1] = "2nd Text Track";
       m_previewView.setEffectOptions( nexEffect.getTitleOptions(effect ID ,aStrings));
       previewView.setEffect( effect ID );
```

Since

version 1.0.0

4.38.2 Member Function Documentation

4.38.2.1 void com.nexstreaming.nexeditorsdk.nexEffectPreviewView.setEffect (String *effect_id*)

This method sets the ID of an effect.

Example code :

```
private nexEffectPreviewView m_previewView;
        m_previewView.setEffect(mSelectedEffectId);
```

Parameters

effect_id	The ID of an effect, as a `String`.

Since

version 1.0.0

See Also

nexEffectLibrary::getClipEffects()
nexEffectLibrary::getTransitionEffects()

4.38.2.2 void com.nexstreaming.nexeditorsdk.nexEffectPreviewView.setEffectOptions (String *effectOptions*)

This method prints text on `nexEffectPreviewView`, by applying the encoded text. Text can be encoded into the options format with the method getTitleOptions(nexEffectOptions).

Example code :

```
String[] aStrings = new String[2];
    aStrings[0] = "1st Text Track";
    aStrings[1] = "2nd Text Track";
    m_previewView.setEffectOptions( nexEffect.getTitleOptions(effect ID ,aStrings));
```

Parameters

effectOptions	The text encoded into the options format.

Since

version 1.0.0

See Also

getTitleOptions(nexEffectOptions)

4.38.2.3 void com.nexstreaming.nexeditorsdk.nexEffectPreviewView.setEffectTime (int *effectTime*)

This method sets the duration of an effect.

Example code :

```
private nexEffectPreviewView m_previewView;
        m_previewView.setEffectTime(3000);
```

Parameters

effectTime	The duration of an effect as an `integer`, in `msec` (milliseconds).

Since

version 1.0.0

4.39 com.nexstreaming.nexeditorsdk.nexEngine Class Reference

Classes

- enum FastPreviewOption
- enum nexErrorCode
- enum nexPlayState

- class OnAutoTrimResultListener
- class OnCompletionListener
- class OnSeekCompletionListener
- class OnSurfaceChangeListener
- class OverlayPreviewBuilder

Public Member Functions

- nexEngine (Context context)
- void setEventHandler (nexEngineListener listener)
- void setOnSurfaceChangeListener (OnSurfaceChangeListener listener)
- int setView (nexEngineView view)
- nexEngineView **getView** ()
- void setProject (nexProject project)
- nexProject getProject ()
- void play ()
- void stop ()
- void exportSaveStop (final nexEngine.OnCompletionListener listener)
- void stop (final nexEngine.OnCompletionListener listener)
- void stopSync ()
- void pause ()
- void resume ()
- void seek (int time)
- void seekIDRorI (int time)
- void seekIDROnly (int time)
- void seekIDROnly (int time, final OnSeekCompletionListener listener)
- int export (String strExportFile, int iEncodeWidth, int iEncodeHeight, int iEncodeBitrate, long iEncodeMaxFileSize, final int iRotate, final int iSamplingRate)
- int **export** (String strExportFile, int iEncodeWidth, int iEncodeHeight, int iEncodeBitrate, long iEncodeMaxFileSize, final int iRotate, final int iSamplingRate, final int fps)
- int export (String strExportFile, int iEncodeWidth, int iEncodeHeight, int iEncodeBitrate, long iEncodeMaxFileSize, final int iRotate)

  This method encodes a project.
- void setScalingFlag2Export (boolean bNeedScaling)
- int exportPause ()
- int exportResume ()
- int checkDirectExport ()
- boolean directExport (String strExportFile, long iEncodeMaxFileSize, long iMaxFileDuration)
- boolean fastPreviewStart (int iStartTime, int iEndTime, int displayWidth, int displayHeight)
- boolean fastPreviewTime (int iSetTime)
- boolean fastPreviewStop ()
- boolean reverseStart (String strSrcClipPath, String strDstClipPath, String strTempClipPath, int iWidth, int iHeight, int iBitrate, long lMaxFileSize, int iStartTime, int iEndTime, int iDecodeMode)
- boolean reverseStop ()
- boolean transcodingStart (String strSrcClipPath, String strDstClipPath, int iWidth, int iHeight, int iDisplayWidth, int iDisplayHeight, int iBitrate, long lMaxFileSize, int iFPS, int iTag)
- boolean transcodingStop ()
- boolean setTotalAudioVolumeWhilePlay (int masterVolume, int slaveVolume)
- boolean setTotalAudioVolumeResetWhilePlay ()
- void setTotalAudioVolumeProject (int masterVolume, int slaveVolume)
- boolean forceMixExport (String strExportFile)
- boolean checkKineMixExport (String srcFilePath, String dstFilePath)
- int checkKineMixExport (boolean dummy)
- boolean checkKineMixExport ()
- boolean KineMixExport (String strExportFile)
- void cancelKineMixExport ()
- int getDuration ()

- int getCurrentPlayTimeTime ()
- void updateProject ()
- void clearProject ()
- void overlayLock (boolean lock)
- boolean getOverlayHitPoint (nexOverlayItem.HitPoint point)
- OverlayPreviewBuilder buildOverlayPreview (int id)
- void fastPreview (FastPreviewOption option, int value)
- void fastPreviewCrop (Rect cropRaw)
- void **fastPreviewEffect** (int cts, boolean swap)
- void clearTrackCache ()
- int[] getIDRSeekTabSync (nexClip clip)
- int addUdta (int type, String data)
- int clearUdta ()
- void updateScreenMode ()
- int autoTrim (String path, boolean IDRunit, int interval, int count, int decodeMode, final OnAutoTrimResultListener listener)
- int autoTrimStop ()
- int getAudioSessionID ()
- boolean set360VideoViewPosition (int angleX, int angleY)
- void set360VideoViewStopPosition (int angleX, int angleY)
- void set360VideoForceNormalView ()

Static Public Member Functions

- static void setLoadListAsync (boolean set)
- static void setExportVideoTrackUUID (boolean on)
- static void prepareSurfaceSetToNull (boolean on)

Static Public Attributes

- static final int retCheckDirectExport_OK = 0
- static final int retCheckDirectExport_InvalidHandle = 1
- static final int retCheckDirectExport_InvalidClipList = 2
- static final int retCheckDirectExport_ClipCountZero = 3
- static final int retCheckDirectExport_InvalidVideoInfo = 4
- static final int retCheckDirectExport_UnmatchedVideoCodec = 5
- static final int retCheckDirectExport_NotStartIFrame = 6
- static final int retCheckDirectExport_HasImageClip = 7
- static final int retCheckDirectExport_HasVideoLayer = 8
- static final int retCheckDirectExport_SetTransitionEffect = 9
- static final int retCheckDirectExport_SetClipEffect = 10
- static final int retCheckDirectExport_EncoderDSIMismatch = 11
- static final int retCheckDirectExport_HasSpeedControl = 12
- static final int retCheckDirectExport_UnsupportedCodec = 13
- static final int retCheckDirectExport_InvalidRotate = 14

4.39.1 Detailed Description

This class previews nexProject with nexClip and nexEffect applied to it, or encodes into MP4 files. This acts as a media player and uses many resources, therefore working with a singleton pattern for the application is suggested.

Example code :

```
protected void onCreate(Bundle savedInstanceState) {
        m_editorView = (nexEngineView)findViewById(R.id.videoView);
        m_editorView.setBlackOut(true);

        mEngine = KMSDKApplication.getApplicationInstance().getEngin();
        mEngine.setView(m_editorView);
        mEngine.setEventHandler(mEditorListener);
        mEngine.setProject(project);
        mEngine.play();
   }
```

See Also

nexProject
nexEngineListener
nexEngineView

Since

version 1.0.0

4.39.2 Constructor & Destructor Documentation

4.39.2.1 com.nexstreaming.nexeditorsdk.nexEngine.nexEngine (Context *context*)

The initial generator of `nexEngine`.

Example code :

```
private nexEngine mEngine;
       mEngin = new nexEngine(mContext);
```

Parameters

context	The android context.

Since

version 1.0.0

4.39.3 Member Function Documentation

4.39.3.1 int com.nexstreaming.nexeditorsdk.nexEngine.addUdta (int *type,* String *data*)

This method adds the desired data in the UDAT box of the MP4 file when exporting.

Parameters

type	The sub-boxes of the asset meta data such as `auth`, `perf`, `gnre`.

data	The user data.

Returns

Zero for success, or a non-zero NexEditor™ error code in the event of a failure.

See Also

clearUdta()

Since

version 1.5.16

4.39.3.2 int com.nexstreaming.nexeditorsdk.nexEngine.autoTrim (String *path,* boolean *IDRunit,* int *interval,* int *count,* int *decodeMode,* final OnAutoTrimResultListener *listener*)

This method make automatic trim data using input parameters `durations` and `count`.

Parameters

path	Source content path (for video only)
interval	Trim duration, in `msec` (milliseconds)
count	Trim count.
listener	listener for AutoTrim result.

Returns

0 always.

Since

version 1.5.15

4.39.3.3 int com.nexstreaming.nexeditorsdk.nexEngine.autoTrimStop ( )

This method stops the Autotrim function.

Returns

Zero for success, or a non-zero NexEditor™ error code in the event of a failure.

4.39.3.4 OverlayPreviewBuilder com.nexstreaming.nexeditorsdk.nexEngine.buildOverlayPreview (int *id*)

This method gets the `OverlayPreviewBuilder` with the given ID.

Parameters

id	The ID to get `OverlayPreviewBuilder`.

Returns

OverlayPreviewBuilder

Since

version 1.5.15

4.39.3.5 void com.nexstreaming.nexeditorsdk.nexEngine.cancelKineMixExport ( )

Deprecated For internal use only. Please do not use.

4.39.3.6 int com.nexstreaming.nexeditorsdk.nexEngine.checkDirectExport ( )

This method checks whether encoding operation is necessary for the export.

Example code :

```
mEngine.checkDirectExport();
```

Returns

One of the value listed above.

Since

version 1.3.4

4.39.3.7 boolean com.nexstreaming.nexeditorsdk.nexEngine.checkKineMixExport (String *srcFilePath,* String *dstFilePath*)

Deprecated For internal use only. Please do not use.

4.39.3.8 int com.nexstreaming.nexeditorsdk.nexEngine.checkKineMixExport (boolean *dummy*)

Deprecated For internal use only. Please do not use.

4.39.3.9 boolean com.nexstreaming.nexeditorsdk.nexEngine.checkKineMixExport ( )

Deprecated For internal use only. Please do not use.

4.39.3.10 void com.nexstreaming.nexeditorsdk.nexEngine.clearProject ( )

This method clears the project loaded on the nexEngine.

Example code :

```
mEngine.clearProject();
```

See Also

setProject(nexProject)

Since

version 1.3.43

4.39.3.11 void com.nexstreaming.nexeditorsdk.nexEngine.clearTrackCache ( )

This method clears the hardware codec and other system sources used by the nexEngine.

Example code :

```
if( mEngin != null) {
            mEngin.stop();
            mEngin.clearTrackCache();
            Log.d(TAG,"clearTrackCache Call!!!");
        }
```

Since

version 1.3.4

4.39.3.12 int com.nexstreaming.nexeditorsdk.nexEngine.clearUdta ( )

This method clears all the user UDAT added using the addUdta(int, String).

Returns

Zero for success, or a non-zero NexEditor™ error code in the event of a failure.

See Also

addUdta(int, String)

Since

version 1.5.16

4.39.3.13 boolean com.nexstreaming.nexeditorsdk.nexEngine.directExport (String *strExportFile,* long *iEncodeMaxFileSize,* long *iMaxFileDuration*)

This method operates an export without encoding.

Example code :

```
mEngine.directExport(mFile.getAbsolutePath(), Long.MAX_VALUE, mProject.
      getTotalTime());
```

Parameters

strExportFile	The path of the file to be exported.
iEncodeMaxFileSize	The maximum file size of the file to be exported.
iMaxFileDuration	The total duration of the original video content.

Returns

`false` if there is no project or failed to checkDirectExport(); otherwise `true`.

Since

version 1.3.4

4.39.3.14 int com.nexstreaming.nexeditorsdk.nexEngine.export (String *strExportFile,* int *iEncodeWidth,* int *iEncodeHeight,* int *iEncodeBitrate,* long *iEncodeMaxFileSize,* final int *iRotate,* final int *iSamplingRate*)

This method encodes a project.

Example code :

```
File file = getExportFile(width, height);
        mEngine.export(file.getAbsolutePath(), width, height, 6 * 1024 * 1024, Long.MAX_VALUE, 0);
```

Parameters

strExportFile	This is a string indicating the path to where the exported file is saved.
iEncodeWidth	The width of the content to be encoded.
iEncodeHeight	The height of the content to be encoded.
iEncodeBitrate	The bitrate of the content to be encoded.
iEncodeMaxFileSize	The maximum size of the content to be encoded. Input the size of available space from the current storage.
iRotate	The rotate tag that matches the iRotate angle in the MP4 file.
iSamplingRate	The Sampling rate tag that matches the sampling rate in the mp4 file.

Exceptions

ProjectNotAttachedException	The case where there is no project created or no clip is created within a project.

Returns

-1 if the SDK is already in export state; 0 if successful.

Since

version 1.5.38

4.39.3.15 int com.nexstreaming.nexeditorsdk.nexEngine.export (String *strExportFile,* int *iEncodeWidth,* int *iEncodeHeight,* int *iEncodeBitrate,* long *iEncodeMaxFileSize,* final int *iRotate*)

This method encodes a project.

Example code :

```
 File file = getExportFile(width, height);
mEngine.export(file.getAbsolutePath(), width, height, 6 * 1024 * 1024, Long.MAX_VALUE, 0);
```

Parameters

strExportFile	This is a string indicating the path to where the exported file is saved.
iEncodeWidth	The width of the content to be encoded.
iEncodeHeight	The height of the content to be encoded.
iEncodeBitrate	The bitrate of the content to be encoded.
iEncodeMaxFileSize	The maximum size of the content to be encoded. Input the size of available space from the current storage.
iRotate	The rotate tag that matches the iRotate angle in the MP4 file.

Exceptions

ProjectNotAttachedException	The case where there is no project created or no clip is created within a project.

Returns

-1 if already export state; 0 if successful.

Since

version 1.0.0

4.39.3.16 int com.nexstreaming.nexeditorsdk.nexEngine.exportPause ( )

This method pauses the application during an export.

Example code :

```
mEngine.exportPause();
```

Exceptions

ProjectNotAttachedException	The case where there is no project created or no clip is created within a project.

Since

version 1.5.19

See Also

exportResume()

Returns

Zero for success, or a non-zero NexEditor™ error code in the event of a failure.

4.39.3.17 int com.nexstreaming.nexeditorsdk.nexEngine.exportResume ( )

This method resumes the export that has been paused.

Example code :

```
mEngine.exportResume();
```

Exceptions

ProjectNotAttachedException	The case where there is no project created or no clip is created within a project.

Since

version 1.5.19

See Also

exportPause()

Returns

Zero for success, or a non-zero NexEditor™ error code in the event of a failure.

4.39.3.18 void com.nexstreaming.nexeditorsdk.nexEngine.exportSaveStop (final nexEngine.OnCompletionListener *listener*)

This method allows to stop and save what's been exported so far while `export()`.

Example code :

```
mEngine.exportSaveStop(new nexEngine.OnCompletionListener() {
            @Override
            public void onComplete(int resultCode) {
                Log.d(TAG, "Save export : "+resultCode);
            }
        });
```

Parameters

listener	OnCompletionListener

See Also

stop()

Since

version 1.3.43

4.39.3.19 void com.nexstreaming.nexeditorsdk.nexEngine.fastPreview (FastPreviewOption *option,* int *value*)

This method allows a color effect to be previewed in real time without skipping, when changing its value.

Parameters

option	The preview options to set for the clip. This will be one of: **normal**, **brightness**, **contrast**, and **saturation**.
value	The value to set to the parameter `option`, in range from -255 to 255.

Example code :

```
mEngine.fastPreview(nexEngine.FastPreviewOption.brightness,value);
```

See Also

FastPreviewOption

Since

version 1.1.0

4.39.3.20 void com.nexstreaming.nexeditorsdk.nexEngine.fastPreviewCrop (Rect *cropRaw*)

This method previews the cropped output screen.

Example code :

```
rect.left = (int) (rect.left / mRatio);
        rect.top = (int) (rect.top / mRatio);
        rect.right = (int) (rect.right / mRatio);
        rect.bottom = (int) (rect.bottom / mRatio);
        mEngine.getProject().getClip(0, true).getCrop().setStartPosition(rect);
        Rect raw = new Rect();
        mEngine.getProject().getClip(0, true).getCrop().getStartPositionRaw(raw);
        mEngine.fastPreviewCrop(raw);
```

Parameters

cropRaw	The value of parameters `getStartPositionRaw()` or `setEndPosition()` of `nexCrop()`.

See Also

getEndPositionRaw(android.graphics.Rect)
getStartPositionRaw(android.graphics.Rect)

Since

version 1.3.43

4.39.3.21 boolean com.nexstreaming.nexeditorsdk.nexEngine.fastPreviewStart (int *iStartTime*, int *iEndTime*, int *displayWidth*, int *displayHeight*)

This method initializes the preview of an overlay item without encoding. This method operates asynchronously.

Example code :

```
mEngine.fastPreviewStart(iStartTime, iStartTime + 6000, 1920, 1080);
```

Parameters

iStartTime	The start time of the preview.
iEndTime	The end time of the preview.
displayWidth	The width of the preview output screen.
displayHeight	The height of the preview output screen.

Returns

Always return `true` and the results will be returned to `onFastPreviewStartDone`.

See Also

fastPreviewTime(int)
fastPreviewStop()

Since

version 1.3.29

4.39.3.22 boolean com.nexstreaming.nexeditorsdk.nexEngine.fastPreviewStop ( )

This method clears an overlay item preview. This method operates asynchronously.

Example code :

```
mEngine.fastPreviewStop();
```

Returns

Always return `true` and the result will be returned to `onFastPreviewStopDone`.

See Also

fastPreviewStart(int, int, int, int)
fastPreviewTime(int)

Since

version 1.3.52

4.39.3.23 boolean com.nexstreaming.nexeditorsdk.nexEngine.fastPreviewTime (int *iSetTime*)

This method sets the display time (timestamp) of an overlay item preview that is initialized by the `fastPreviewStart`. This method operates asynchronously.

Example code :

```
int iTime = progress + mStartTime;
        mEngine.fastPreviewTime(iTime);
```

Parameters

iSetTime	The desired display time (timestamp) of the overlay item preview.

Returns

Always return `true` and the result will be returned to `onFastPreviewTimeDone`.

See Also

fastPreviewStart(int, int, int, int)
fastPreviewStop()

Since

version 1.3.29

4.39.3.24 boolean com.nexstreaming.nexeditorsdk.nexEngine.forceMixExport (String *strExportFile*)

This method allows to export a project without using any encoder when all the clips are referencing a single source file. Exporting with this method is faster because there is no use of encoder. This method must be called when `nexEngine` is in the `Stop` state.

Example code :

```
if(mEngine.forceMixExport(mFile.getAbsolutePath()) == false)
{
    Log.e(TAG, "forceMixExport error");
}
```

Parameters

strExportFile	The path where the output files are saved.

Returns

Exported output video file; otherwise `FALSE` if the project is not in the suitable condition to be used by this method. For example, when there is an image clip in the project or when the clips in the project are not referencing a single source file.

Since

version 1.3.43

4.39.3.25 int com.nexstreaming.nexeditorsdk.nexEngine.getAudioSessionID ( )

For internal use only. Please do not use.

Returns

Since

1.5.42

4.39.3.26 int com.nexstreaming.nexeditorsdk.nexEngine.getCurrentPlayTimeTime ( )

This method gets the current play time, on the timeline, of a clip.

Example code :

```
int currentplaytime = mEngine.getCurrentPlayTimeTime();
```

Returns

The amount of time played already since the start time of a clip.

See Also

setEventHandler(nexEngineListener)

Since

version 1.0.1

4.39.3.27 int com.nexstreaming.nexeditorsdk.nexEngine.getDuration ( )

This method gets the duration of a project. To get the exact duration value, this method should be used after calling play() then nexEngineListener#onPlayStart().

The method nexProject#getTotalTime() of the class `nexProject` can be used instead of this method.

Example code :

```
int duration = mEngine.getDuration();
```

Returns

The duration of the project in `msec` (milliseconds); 0 if no clip in the project.

Exceptions

ProjectNotAttached-Exception	in the case where there is no project created or no clip is created within a project.

Since

version 1.0.0

4.39.3.28 int [ ] com.nexstreaming.nexeditorsdk.nexEngine.getIDRSeekTabSync (nexClip *clip*)

This method gets the IDR-frame locations of a video file in an `array`. This method operates slower than `getSeekPointsSync()`. If the length of the content gets too long, ANR may occur.

Example code :

```
int seektab[] = mEngine.getIDRSeekTabSync(mProject.getClip(0, true).getPath());
```

Parameters

clip	The path where the video content, to retrieve IDR-frames, is saved.

Returns

The time location of the IDR-frames in `array`.

See Also

nexClip::getSeekPointsSync()

Since

version 1.3.43

4.39.3.29 boolean com.nexstreaming.nexeditorsdk.nexEngine.getOverlayHitPoint (nexOverlayItem.HitPoint *point*)

This method tells whether any overlay item has been selected from the touched location on the current screen.

Parameters

point	Touched location on the screen.

Returns

`true` if any overlay item has been selected.

Since

version 1.5.15

See Also

HitPoint

4.39.3.30 nexProject com.nexstreaming.nexeditorsdk.nexEngine.getProject ( )

This method gets a project currently set to `nexEngine`.

Example code :

```
mEngine.getProject().getClip(0,true).setBrightness(value);
```

Returns

The project set to `nexEngine`.

Exceptions

ProjectNotAttachedException	

Since

version 1.0.0

See Also

setProject(nexProject)

4.39.3.31 boolean com.nexstreaming.nexeditorsdk.nexEngine.KineMixExport (String *strExportFile*)

Deprecated For internal use only. Please do not use.

4.39.3.32 void com.nexstreaming.nexeditorsdk.nexEngine.overlayLock (boolean *lock*)

This method sets whether to lock or unlock an overlay clip.

Example code :

```
mEngine.overlayLock(true);
```

Note

This method is for the debugging purpose. Please do not use.

Parameters

lock	Set to `TRUE` to lock; otherwise set to `FALSE` to unlock the overlay clip.

4.39.3.33 void com.nexstreaming.nexeditorsdk.nexEngine.pause ( )

This method will pause the content if the project is already set.

Example code :

```
if( !isSeeking ){
            mEngine.pause();
            isSeeking = true;
        }
```

Exceptions

ProjectNotAttachedException	The case where there is no project created or no clip is created within a project.

Since

version 1.0.0

See Also

resume()
play()
stop()
seek(int)

4.39.3.34 void com.nexstreaming.nexeditorsdk.nexEngine.play ( )

This method will play content from the beginning if the project is already set.

Example code :

```
mEngine.setProject(project);
        mEngine.play();
```

Exceptions

ProjectNotAttachedException	The case where there is no project created or no clip is created within a project.

Since

version 1.0.0

See Also

pause()
resume()
stop()
seek(int)

4.39.3.35 static void com.nexstreaming.nexeditorsdk.nexEngine.prepareSurfaceSetToNull (boolean *on*) [static]

For internal use only. Please do not use.

Parameters

on	

Since

1.5.42

4.39.3.36 void com.nexstreaming.nexeditorsdk.nexEngine.resume ( )

This method will resume the content if the project is already set.

Example code :

```
if( mCurrentPlayTime > 0 ){
            mEngine.seek(mCurrentPlayTime);
        }
        mEngine.resume();
```

Exceptions

ProjectNotAttachedException	The case where there is no project created or no clip is created within a project.

Since

version 1.0.0

See Also

pause()
play()
stop()
seek(int)

4.39.3.37 boolean com.nexstreaming.nexeditorsdk.nexEngine.reverseStart (String *strSrcClipPath*, String *strDstClipPath*, String *strTempClipPath*, int *iWidth*, int *iHeight*, int *iBitrate*, long *lMaxFileSize*, int *iStartTime*, int *iEndTime*, int *iDecodeMode*)

This method allows to reverse play video content using the information set with the parameters. This method operates asynchronously.

Example code :

```
String srcPath = mListFilePath.get(0);
        mDstFilePath = getReverseDstFilePath();
        String temppath = getReverseTempFilePath();
        int width = mProject.getClip(0, true).getWidth();
        int height = mProject.getClip(0, true).getHeight();
        int bitrate = mProject.getClip(0, true).getVideoBitrate();
        long maxFileSize = Long.MAX_VALUE;
        mEngine.reverseStart(srcPath, mDstFilePath, temppath, width, height, bitrate, maxFileSize, mStartT,
       mEndT);
```

Parameters

strSrcClipPath	The location where the original video content is stored.
strDstClipPath	The location where the output file is saved.
strTempClipPath	The path temporarily needed for operating this method.
iWidth	The width of the output content.
iHeight	The height of the output content.
iBitrate	The bitrate of the output content.
lMaxFileSize	The maximum file size for the output content.
iStartTime	The timestamp of the original video content where the reverse playback will start from.
iEndTime	The timestamp of the original video content where the reverse playback will end.

Returns

`true` if the task for `reverseStart` starts, otherwise `false`.

See Also

reverseStop()

Since

version 1.3.32

4.39.3.38 boolean com.nexstreaming.nexeditorsdk.nexEngine.reverseStop ( )

This method stops the reverse play of video content that's currently reverse playing.

Example code :

```
mEngine.reverseStop();
```

Returns

`true` if successful, otherwise `false`.

See Also

reverseStart(String, String, String, int, int, int, long, int, int, int)

Since

version 1.3.32

4.39.3.39 void com.nexstreaming.nexeditorsdk.nexEngine.seek (int *time*)

This method seeks for content if the project is already set. For this method to work properly, methods like `stop()` or `pause()` should be called before. Otherwise, `seek()` method will fail and an event listener `onSetTimeFail()` will occur.

Example code :

```
if( mCurrentPlayTime > 0 ){
            mEngine.seek(mCurrentPlayTime);
        }
        mEngine.resume();
```

Parameters

time	The seek time in `msec` (milliseconds).

Exceptions

ProjectNotAttachedException	The case where there is no project created or no clip is created within a project.

Since

version 1.0.0

See Also

pause()
resume()
play()
stop()

4.39.3.40 void com.nexstreaming.nexeditorsdk.nexEngine.seekIDROnly (int *time*)

This method seeks for the closest IDR-frame, before the input seek time value by the parameter `time`, of video content. Therefore, the actual seek result shown on the preview might be different from the input seek time.

Example code :

```
mEngine.seekIDROnly(seekTime);
```

Parameters

time	The seek time in `msec` millisecond.

See Also

seekIDROnly(int, OnSeekCompletionListener)
seekIDRorI(int)
seek(int)

Since

version 1.3.43

4.39.3.41 void com.nexstreaming.nexeditorsdk.nexEngine.seekIDROnly (int *time,* final OnSeekCompletionListener *listener*)

This method seeks for the closest IDR-frame, before the input seek time value by the parameter `time`, of video content. Therefore, the actual seek result shown on the preview might be different from the input seek time. To get the actual seek frame, use the parameter `listener` to get the value from parameter `iFrameTime` of `onSeekComplete(int time, int iFrameTime).`

Example code :

```
mEngine.seekIDROnly(seek, new nexEngine.OnSeekCompletionListener() {
            @Override
            public void onSeekComplete(int resultCode, int seekTime, int movedTime) {
                if( resultCode == 0 ){
                    Log.d(TAG,"onSeekComplete: seekTime="+seekTime+", movedTime="+movedTime);
                }
            }
        });
```

Parameters

time	The seek time in `msec` millisecond.
listener	OnSeekCompletionListener. Notifies when the `seek` has completed and returns the actual seek location of an IDR-frame.

See Also

seekIDROnly(int)
seekIDRorI(int)
seek(int)

Since

version 1.3.43

4.39.3.42 void com.nexstreaming.nexeditorsdk.nexEngine.seekIDRorI (int *time*)

This method seeks for the closest I-frame, before the input seek time value by the parameter `time`, of video content. Therefore, the actual seek result shown on the preview might be different from the input seek time.

Example code :

```
if(mSeekMode == 1) {
            mEngine.seek(ms);
        } else {
            mEngine.seekIDRorI(ms);
        }
```

Parameters

time	The seek time in `msec` millisecond.

See Also

seekIDROnly(int, OnSeekCompletionListener)
seekIDROnly(int)
seek(int)

Since

version 1.3.43

4.39.3.43 void com.nexstreaming.nexeditorsdk.nexEngine.set360VideoForceNormalView ( )

In order to play regular video content from a screen on 360 video mode, the screen needs to swtich to regular view mode. This forces 360 mode to switch to normal view mode.

Since

1.7.0

4.39.3.44 boolean com.nexstreaming.nexeditorsdk.nexEngine.set360VideoViewPosition (int *angleX*, int *angleY*)

This moves the view position when playing 360 video content.

Parameters

angleX	-360 ∼ 360
angleY	-360 ∼ 360

Returns

Since

1.5.42

4.39.3.45 void com.nexstreaming.nexeditorsdk.nexEngine.set360VideoViewStopPosition (int *angleX*, int *angleY*)

This moves the view position when 360 video content is paused during playback.

Parameters

angleX	-360 ∼ 360
angleY	-360 ∼ 360

Since

1.5.42

4.39.3.46 void com.nexstreaming.nexeditorsdk.nexEngine.setEventHandler (nexEngineListener *listener*)

This method registers nexEngineListener of nexEngine.

For Example :

```
 mEngine.setEventHandler( new nexEngineListener(){
        @literal @Override
         public void onStateChange(int oldState, int newState) {

         }
         ...............
         @literal @Override
         public void onEncodingProgress(int percent) {

         }
};);
```

Parameters

listener	nexEngineListener

Since

version 1.0.0

4.39.3.47 static void com.nexstreaming.nexeditorsdk.nexEngine.setExportVideoTrackUUID (boolean *on*) [static]

This method sets the UUID to the mp4 file of the export video. If the original clip video shows a UUID during exporting, that means it was taken by a 360 camcorder.

If there is no UUID, the method will not add anything to the export result file.

Parameters

on	The availability of the UUID. • **TRUE** Sets the UUID of the original clip video taken by a 360 camcorder to the export video. • **FALSE** Do not check the UUID.

Since

version 1.5.25

4.39.3.48 static void com.nexstreaming.nexeditorsdk.nexEngine.setLoadListAsync (boolean *set*) [static]

This method allows to choose whether to operate `updateProject()` asynchronously or synchronously. When this method operates as default synchronously, `updateProject()` must be called after all commands are completed and the engine is in the state of `Stop`.

Example code :

```
mEngin.setLoadListAsync(true);
```

Parameters

set	`TRUE` to operate asynchronously, or `FALSE` to operate synchronously.

Since

version 1.3.43

4.39.3.49 void com.nexstreaming.nexeditorsdk.nexEngine.setOnSurfaceChangeListener (OnSurfaceChangeListener *listener*)

This method registers a listener to receive an event when the surface changes.

Example code :

```
mEngine.setOnSurfaceChangeListener(new nexEngine.OnSurfaceChangeListener() {
        @Override
        public void onSurfaceChanged() {
            if( !mSeek ){
                mSeek = true;
                mEngine.updateProject();
                mEngine.seek(1000);
                }
            }
        });
```

Parameters

listener	A listener made by inheriting `nexEngine.OnSurfaceChangeListener`.

Since

version 1.0.0

4.39.3.50 void com.nexstreaming.nexeditorsdk.nexEngine.setProject (nexProject *project*)

This method sets a project to be played and encoded.

Example code :

```
nexClip clip=nexClip.getSupportedClip(m_listfilepath.get(0));
        if( clip != null ){
            project.add(clip);
        }
        mEngine.setProject(project);
```

Parameters

project	The project to set.

Since

version 1.0.0

See Also

getProject()

4.39.3.51 void com.nexstreaming.nexeditorsdk.nexEngine.setScalingFlag2Export (boolean *bNeedScaling*)

This is an API that can scale the sticker size, which is normally set to 720p.

Parameters

bNeedScaling	Set to `TRUE` to turn on the scaling motion; otherwise, `FALSE`.

Note

Please note that scaling may cause a delay in exporting time.

Since

version 1.5.25

4.39.3.52 void com.nexstreaming.nexeditorsdk.nexEngine.setTotalAudioVolumeProject (int *masterVolume,* int *slaveVolume*)

This method sets all the volumes in the project.

Parameters

masterVolume	The video clip volume.
slaveVolume	The audio clip volume.

Since

version 1.5.15

4.39.3.53 boolean com.nexstreaming.nexeditorsdk.nexEngine.setTotalAudioVolumeResetWhilePlay ( )

This method sets the volume to default while preview is playing.

Returns

`TRUE` if successful, otherwise `FALSE`.

See Also

setTotalAudioVolumeWhilePlay(int, int)

Since

version 1.5.15

4.39.3.54 boolean com.nexstreaming.nexeditorsdk.nexEngine.setTotalAudioVolumeWhilePlay (int *masterVolume,* int *slaveVolume*)

This method controls the volume while preview is playing.

Parameters

masterVolume	The video clip volume.
slaveVolume	The audio clip volume.

Returns

`TRUE` if successful, otherwise `FALSE`.

See Also

setTotalAudioVolumeResetWhilePlay()

Since

version 1.5.15

4.39.3.55 int com.nexstreaming.nexeditorsdk.nexEngine.setView (nexEngineView *view*)

This method registers a view of an application.

Example code :

```
private nexEngineView m_editorView;

        m_editorView = (nexEngineView)findViewById(R.id.videoView_bgmtest);
        m_editorView.setBlackOut(true);
        mEngine = KMSDKApplication.getApplicationInstance().getEngin();
        mEngine.setView(m_editorView);
```

Parameters

view	The surface view.

Since

version 1.0.0

4.39.3.56 void com.nexstreaming.nexeditorsdk.nexEngine.stop ( )

This method will stop the content if the project is already set.

Example code :

```
if( mExportStat ) {
            mEngine.stop();
            mExportStat = false;
        }
```

Exceptions

ProjectNotAttachedException	The case where there is no project created or no clip is created within a project.

Since

version 1.0.0

See Also

pause()
resume()
play()
seek(int)

4.39.3.57 void com.nexstreaming.nexeditorsdk.nexEngine.stop (final nexEngine.OnCompletionListener *listener*)

This method returns the completion time of `Stop()` using `OnCompletionListener`.

Example code :

```
mEngine.stop(new nexEngine.OnCompletionListener() {
            @Override
            public void onComplete(int resultCode) {
                mEngine.getProject().getClip(0,true).getClipEffect().setEffectShowTime(mStartTime, mEndTime
      );
                mEngine.play();
            }
        });
```

Parameters

listener	OnCompletionListener

See Also

play()
resume()
#export(String, int, int, int, long, int)

Since

version 1.3.43

4.39.3.58 void com.nexstreaming.nexeditorsdk.nexEngine.stopSync ( )

Deprecated For internal use only. Please do not use.

4.39.3.59 boolean com.nexstreaming.nexeditorsdk.nexEngine.transcodingStart (String *strSrcClipPath,* String *strDstClipPath,* int *iWidth,* int *iHeight,* int *iDisplayWidth,* int *iDisplayHeight,* int *iBitrate,* long *lMaxFileSize,* int *iFPS,* int *iTag*)

This method starts transcoding video content using the information set with the parameters.

Parameters

strSrcClipPath	The location where the original video content is stored.
strDstClipPath	The location where the output file is saved.
iWidth	The width of the output content.
iHeight	The height of the output content.
iBitrate	The bitrate of the output content.
lMaxFileSize	The maximum file size for the output content.
iFPS	The FPS of the output content.
iTag	The Rotate flag.

Returns

`true` if successful, otherwise `false`.

See Also

transcodingStop()

Since

version 1.5.15

4.39.3.60 boolean com.nexstreaming.nexeditorsdk.nexEngine.transcodingStop ( )

This method stops transcoding video content that's currently transcoding.

Returns

`true` if successful, otherwise `false`.

See Also

transcodingStart(String, String, int, int, int, int, int, long, int, int)

Since

version 1.5.15

4.39.3.61 void com.nexstreaming.nexeditorsdk.nexEngine.updateProject ( )

This method updates the project when there is any change made to the project settings.

Example code :

```
mEngine = KMSDKApplication.getApplicationInstance().getEngin();
        mEngine.setView(m_editorView);
        mEngine.setProject(project);
        mEngine.updateProject();
```

Since

version 1.0.1

4.39.3.62 void com.nexstreaming.nexeditorsdk.nexEngine.updateScreenMode ( )

This method updates the screen mode with the values set to `getScreenMode` and `getAspectRatioMode`.

Example code :

```
nexApplicationConfig.setScreenMode(nexApplicationConfig.kScreenMode_horizonDual
      );
        mEngine.updateScreenMode();
```

See Also

nexApplicationConfig::getScreenMode()
nexApplicationConfig::getAspectRatioMode()

Since

version 1.3.52

4.39.4 Member Data Documentation

4.39.4.1 final int com.nexstreaming.nexeditorsdk.nexEngine.retCheckDirectExport_ClipCountZero = 3 [static]

This is a possible return value for checkDirectExport() when there is no clip in the project.

Since

version 1.3.4

4.39.4.2 final int com.nexstreaming.nexeditorsdk.nexEngine.retCheckDirectExport_EncoderDSIMismatch = 11 [static]

This is a possible return value for checkDirectExport() when DSI is not supported by the encoder.

Since

version 1.3.43

4.39.4.3 final int com.nexstreaming.nexeditorsdk.nexEngine.retCheckDirectExport_HasImageClip = 7 [static]

This is a possible return value for checkDirectExport() when there is an image in the project.

Since

version 1.3.4

4.39.4.4 final int com.nexstreaming.nexeditorsdk.nexEngine.retCheckDirectExport_HasSpeedControl = 12 [static]

This is a possible return value for checkDirectExport() when a clip's speed has been adjusted.

Since

version 1.3.43

4.39.4.5 final int com.nexstreaming.nexeditorsdk.nexEngine.retCheckDirectExport_HasVideoLayer = 8 [static]

This is a possible return value for checkDirectExport() when there is an overlay image in the project.

Since

version 1.3.4

4.39.4.6 final int com.nexstreaming.nexeditorsdk.nexEngine.retCheckDirectExport_InvalidClipList = 2 [static]

This is a possible return value for checkDirectExport() when there is an invalid clip in the project.

Since

version 1.3.4

4.39.4.7 final int com.nexstreaming.nexeditorsdk.nexEngine.retCheckDirectExport_InvalidHandle = 1 [static]

This is a possible return value for checkDirectExport() when there is no engine handle available.

Since

version 1.3.4

4.39.4.8 final int com.nexstreaming.nexeditorsdk.nexEngine.retCheckDirectExport_InvalidRotate = 14 [static]

This is a possible return value for checkDirectExport() when the rotate information of clips is different.

Since

version 1.3.43

4.39.4.9 final int com.nexstreaming.nexeditorsdk.nexEngine.retCheckDirectExport_InvalidVideoInfo = 4 [static]

This is a possible return value for checkDirectExport() when there is an invalid video file in the project.

Since

version 1.3.4

4.39.4.10 final int com.nexstreaming.nexeditorsdk.nexEngine.retCheckDirectExport_NotStartIFrame = 6 [static]

This is a possible return value for checkDirectExport() when a clip is trimmed into two separate clips, the first frame of the second clip is not an I-frame.

Since

version 1.3.4

4.39.4.11 final int com.nexstreaming.nexeditorsdk.nexEngine.retCheckDirectExport_OK = 0 [static]

This is a possible return value for checkDirectExport() when the export is possible.

Since

version 1.3.4

4.39.4.12 final int com.nexstreaming.nexeditorsdk.nexEngine.retCheckDirectExport_SetClipEffect = 10 [static]

This is a possible return value for checkDirectExport() when there is a clip effect in the project.

Since

version 1.3.4

4.39.4.13 final int com.nexstreaming.nexeditorsdk.nexEngine.retCheckDirectExport_SetTransitionEffect = 9 [static]

This is a possible return value for checkDirectExport() when there is a transition effect in the project.

Since

version 1.3.4

4.39.4.14 final int com.nexstreaming.nexeditorsdk.nexEngine.retCheckDirectExport_UnmatchedVideoCodec = 5 [static]

This is a possible return value for checkDirectExport() when there is a video with unsupported file format in the project.

Since

version 1.3.4

4.39.4.15 final int com.nexstreaming.nexeditorsdk.nexEngine.retCheckDirectExport_UnsupportedCodec = 13 [static]

This is a possible return value for checkDirectExport() when the codec is not supported by the checkDirectExport().

Since

version 1.3.43

4.40 com.nexstreaming.nexeditorsdk.nexEngineListener Interface Reference

Public Member Functions

- void onStateChange (int oldState, int newState)
- void onTimeChange (int currentTime)
- void onSetTimeDone (int currentTime)
- void onSetTimeFail (int err)
- void onSetTimeIgnored ()
- void onEncodingDone (boolean iserror, int result)
- void onPlayEnd ()
- void onPlayFail (int err, int iClipID)
- void onPlayStart ()
- void onClipInfoDone ()
- void onSeekStateChanged (boolean isSeeking)
- void onEncodingProgress (int percent)
- void onCheckDirectExport (int result)
- void onProgressThumbnailCaching (int progress, int maxValue)
- void onFastPreviewStartDone (int err, int startTime, int endTime)
- void onFastPreviewStopDone (int err)
- void onFastPreviewTimeDone (int err)
- void onPreviewPeakMeter (int iCts, int iPeakMeterValue)

4.40.1 Detailed Description

The application must implement this interface, from nexEngine nexEngine#setEventHandler(nexEngineListener), in order to receive events from NexEditor™ SDK.

State Value

- NONE(0)
- IDLE(1)
- RUN(2)
- RECORD(3)

Error Value

- NONE(0),
- GENERAL(1),
- UNKNOWN(2),
- NO_ACTION(3),
- INVALID_INFO(4),
- INVALID_STATE(5),
- VERSION_MISMATCH(6),
- CREATE_FAILED(7),
- MEMALLOC_FAILED(8),
- ARGUMENT_FAILED(9),
- NOT_ENOUGH_NEMORY(10),
- EVENTHANDLER(11),
- FILE_IO_FAILED(12),
- FILE_INVALID_SYNTAX(13),
- FILEREADER_CREATE_FAIL(14),
- FILEWRITER_CREATE_FAIL(15),
- AUDIORESAMPLER_CREATE_FAIL(16),
- UNSUPPORT_FORMAT(17),
- FILEREADER_FAILED(18),
- PLAYSTART_FAILED(19),
- PLAYSTOP_FAILED(20),
- PROJECT_NOT_CREATE(21),
- PROJECT_NOT_OPEN(22),
- CODEC_INIT(23),
- RENDERER_INIT(24),
- THEMESET_CREATE_FAIL(25),

- ADD_CLIP_FAIL(26),
- ENCODE_VIDEO_FAIL(27),
- INPROGRESS_GETCLIPINFO(28),
- THUMBNAIL_BUSY(29),
- UNSUPPORT_MIN_DURATION(30),
- UNSUPPORT_MAX_RESOLUTION(31),
- UNSUPPORT_MIN_RESOLUTION(32),
- UNSUPPORT_VIDEIO_PROFILE(33),
- UNSUPPORT_VIDEO_LEVEL(34),
- UNSUPPORT_VIDEO_FPS(35),
- TRANSCODING_BUSY(36),
- TRANSCODING_NOT_SUPPORTED_FORMAT(37),
- TRANSCODING_USER_CANCEL(38),
- TRANSCODING_NOT_ENOUGHT_DISK_SPACE(39),
- TRANSCODING_CODEC_FAILED(40),
- EXPORT_WRITER_INVAILED_HANDLE(41),
- EXPORT_WRITER_INIT_FAIL(42),
- EXPORT_WRITER_START_FAIL(43),
- EXPORT_AUDIO_DEC_INIT_FAIL(44),
- EXPORT_VIDEO_DEC_INIT_FAIL(45),
- EXPORT_VIDEO_ENC_FAIL(46),
- EXPORT_VIDEO_RENDER_INIT_FAIL(47),
- EXPORT_NOT_ENOUGHT_DISK_SPACE(48),
- UNSUPPORT_AUDIO_PROFILE(49),
- THUMBNAIL_INIT_FAIL(50),
- UNSUPPORT_AUDIO_CODEC(51),
- UNSUPPORT_VIDEO_CODEC(52),
- HIGHLIGHT_FILEREADER_INIT_ERROR(53),
- HIGHLIGHT_TOO_SHORT_CONTENTS(54),
- HIGHLIGHT_CODEC_INIT_ERROR(55),
- HIGHLIGHT_CODEC_DECODE_ERROR(56),
- HIGHLIGHT_RENDER_INIT_ERROR(57),
- HIGHLIGHT_WRITER_INIT_ERROR(58),
- HIGHLIGHT_WRITER_WRITE_ERROR(59),
- HIGHLIGHT_GET_INDEX_ERROR(60),
- HIGHLIGHT_USER_CANCEL(61),
- GETCLIPINFO_USER_CANCEL(62),
- DIRECTEXPORT_CLIPLIST_ERROR (63),
- DIRECTEXPORT_CHECK_ERROR(64),

- DIRECTEXPORT_FILEREADER_INIT_ERROR(65),
- DIRECTEXPORT_FILEWRITER_INIT_ERROR(66),
- DIRECTEXPORT_DEC_INIT_ERROR(67),
- DIRECTEXPORT_DEC_INIT_SURFACE_ERROR(68),
- DIRECTEXPORT_DEC_DECODE_ERROR(69),
- DIRECTEXPORT_ENC_INIT_ERROR(70),
- DIRECTEXPORT_ENC_ENCODE_ERROR(71),
- DIRECTEXPORT_ENC_INPUT_SURFACE_ERROR(72),
- DIRECTEXPORT_ENC_FUNCTION_ERROR(73),
- DIRECTEXPORT_ENC_DSI_DIFF_ERROR(74),
- DIRECTEXPORT_ENC_FRAME_CONVERT_ERROR(75),
- DIRECTEXPORT_RENDER_INIT_ERROR(76),
- DIRECTEXPORT_WRITER_WRITE_ERROR(77),
- DIRECTEXPORT_WRITER_UNKNOWN_ERROR(78),
- FASTPREVIEW_USER_CANCEL(79),
- FASTPREVIEW_CLIPLIST_ERROR(80),
- FASTPREVIEW_FIND_CLIP_ERROR(81),
- FASTPREVIEW_FIND_READER_ERROR(82),
- FASTPREVIEW_VIDEO_RENDERER_ERROR(83),
- FASTPREVIEW_DEC_INIT_SURFACE_ERROR(84),
- HW_NOT_ENOUGH_MEMORY(85),
- EXPORT_USER_CANCEL(86),
- FASTPREVIEW_DEC_INIT_ERROR(87),
- FASTPREVIEW_FILEREADER_INIT_ERROR(88),
- FASTPREVIEW_TIME_ERROR(89),
- FASTPREVIEW_RENDER_INIT_ERROR(90),
- FASTPREVIEW_OUTPUTSURFACE_INIT_ERROR(91),
- FASTPREVIEW_FASTPREVIEW_BUSY(92),
- CODEC_DECODE(93),
- RENDERER_AUDIO(94);

Since

version 1.0.0

4.40.2 Member Function Documentation

4.40.2.1 void com.nexstreaming.nexeditorsdk.nexEngineListener.onCheckDirectExport (int *result*)

This method returns the result of nexEngine#checkDirectExport().

Parameters

result	The value input as the parameter `result` of the method nexEngine#checkDirectExport().

Since

1.3.17

4.40.2.2 void com.nexstreaming.nexeditorsdk.nexEngineListener.onClipInfoDone ( )

Deprecated For internal use only. Please do not use.

4.40.2.3 void com.nexstreaming.nexeditorsdk.nexEngineListener.onEncodingDone (boolean *iserror,* int *result*)

This method is called when nexEngine#export(String, int, int, int, long, int) operation has completed.

Parameters

iserror	
result	

Since

version 1.0.0

4.40.2.4 void com.nexstreaming.nexeditorsdk.nexEngineListener.onEncodingProgress (int *percent*)

This method returns the current process percentage after running nexEngine#export(String, int, int, int, long, int).

Parameters

percent	The current export process in percentage.

Since

version 1.0.0

4.40.2.5 void com.nexstreaming.nexeditorsdk.nexEngineListener.onFastPreviewStartDone (int *err,* int *startTime,* int *endTime*)

This method returns the status of `fastPreviewStart` function.

Parameters

err	An error code.
startTime	The start time of the fast preview.
endTime	The end time of the fast preview.

Since

version 1.3.21

4.40.2.6 void com.nexstreaming.nexeditorsdk.nexEngineListener.onFastPreviewStopDone (int *err*)

This method returns the status of `fastPreviewStop` function.

Parameters

err	An error code.

Since

version 1.3.21

4.40.2.7 void com.nexstreaming.nexeditorsdk.nexEngineListener.onFastPreviewTimeDone (int *err*)

This method returns the status of `fastPreviewTime` function.

Parameters

err	An error code.

Since

version 1.3.21

4.40.2.8 void com.nexstreaming.nexeditorsdk.nexEngineListener.onPlayEnd ( )

This method is called when nexEngine#play() ends.

Since

version 1.0.0

4.40.2.9 void com.nexstreaming.nexeditorsdk.nexEngineListener.onPlayFail (int *err,* int *iClipID*)

This method is called when nexEngine#play() fails.

Parameters

err	
iClipID	

Since

version 1.0.0

4.40.2.10 void com.nexstreaming.nexeditorsdk.nexEngineListener.onPlayStart ( )

This method is called when nexEngine#play() starts.

Since

version 1.0.0

4.40.2.11 void com.nexstreaming.nexeditorsdk.nexEngineListener.onPreviewPeakMeter (int *iCts,* int *iPeakMeterValue*)

This method returns the Peak value of Audio in preview.

Parameters

iCts	
iPeakMeterValue	

Since

version

4.40.2.12 void com.nexstreaming.nexeditorsdk.nexEngineListener.onProgressThumbnailCaching (int *progress,* int *maxValue*)

This method returns the current process percentage when retrieving thumbnails.

Parameters

progress	The current retrieving process in percentage.
maxValue	The maximum number of thumbnails to be retrieved by this method.

Since

1.3.21

4.40.2.13 void com.nexstreaming.nexeditorsdk.nexEngineListener.onSeekStateChanged (boolean *isSeeking*)

This method notifies the application if the `nexEngine` is currently seeking.

Parameters

isSeeking	set to `TRUE` if seeking; `FALSE` if seeking has completed.

Since

version 1.0.0

4.40.2.14 void com.nexstreaming.nexeditorsdk.nexEngineListener.onSetTimeDone (int *currentTime*)

This method is called when nexEngine#seek(int) is successful.

Parameters

currentTime	

Since

version 1.0.0

4.40.2.15 void com.nexstreaming.nexeditorsdk.nexEngineListener.onSetTimeFail (int *err*)

This method is called when nexEngine#seek(int) has failed.

Parameters

err	

Since

version 1.0.0

4.40.2.16 void com.nexstreaming.nexeditorsdk.nexEngineListener.onSetTimeIgnored ( )

This method is called when ignored by nexEngine#seek(int).

Since

version 1.0.0

4.40.2.17 void com.nexstreaming.nexeditorsdk.nexEngineListener.onStateChange (int *oldState,* int *newState*)

This method is called when there is a change of status.

Parameters

oldState	
newState	

Since

version 1.0.0

4.40.2.18 void com.nexstreaming.nexeditorsdk.nexEngineListener.onTimeChange (int *currentTime*)

This method is called when time changes while nexEngine#play().

Parameters

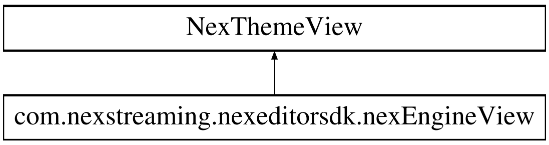
currentTime	

Since

version 1.0.0

4.41 com.nexstreaming.nexeditorsdk.nexEngineView Class Reference

Inheritance diagram for com.nexstreaming.nexeditorsdk.nexEngineView:



Public Member Functions

- nexEngineView (Context context)
- nexEngineView (Context context, AttributeSet attrs)
- void setBlackOut (boolean blackOut)

4.41.1 Detailed Description

This class previews projects if there is any project in `nexEngine`.

Example code:

Layout XML

```
<com.nexstreaming.nexeditorsdk.nexEngineView
            android:id="@+id/videoView"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_gravity="center"/>
```

Class sample code

```
nexEngineView  m_editorView = (nexEngineView)findViewById(R.id.XXXXX);
            m_editorView.setBlackOut(true);
            mEngine.setView(m_editorView);
```

Since

version 1.0.0

4.41.2 Constructor & Destructor Documentation

4.41.2.1 com.nexstreaming.nexeditorsdk.nexEngineView.nexEngineView (Context *context*)

Constructor for `nexEngineView`.

Parameters

context	The application context.

Since

version 1.1.0

4.41.2.2 com.nexstreaming.nexeditorsdk.nexEngineView.nexEngineView (Context *context*, AttributeSet *attrs*)

Constructor for `nexEngineView`.

Parameters

context	The application context.
attrs	The set of attributes.

Since

version 1.1.0

4.41.3 Member Function Documentation

4.41.3.1 void com.nexstreaming.nexeditorsdk.nexEngineView.setBlackOut (boolean *blackOut*)

This method black outs the screen to clear out the surface for the next content.

When the application is reopened, the screen might still be showing an image from the previous content. To prevent this, this method should be used before or after playing a content.

Parameters

blackOut	Set to TRUE to enable black out, set to FALSE to disable black out.

Since

version 1.0.0

Deprecated For internal use only. Please do not use.

4.42 com.nexstreaming.nexeditorsdk.nexEngine.nexErrorCode Enum Reference

Public Member Functions

- **nexErrorCode** (int value)
- int **getValue** ()

Static Public Member Functions

- static nexErrorCode **fromValue** (int value)

Public Attributes

- **NONE** =(0)
- **GENERAL** =(1)
- **UNKNOWN** =(2)
- **NO_ACTION** =(3)
- **INVALID_INFO** =(4)
- **INVALID_STATE** =(5)
- **VERSION_MISMATCH** =(6)
- **CREATE_FAILED** =(7)
- **MEMALLOC_FAILED** =(8)
- **ARGUMENT_FAILED** =(9)
- **NOT_ENOUGH_NEMORY** =(10)
- **EVENTHANDLER** =(11)
- **FILE_IO_FAILED** =(12)
- **FILE_INVALID_SYNTAX** =(13)
- **FILEREADER_CREATE_FAIL** =(14)
- **FILEWRITER_CREATE_FAIL** =(15)
- **AUDIORESAMPLER_CREATE_FAIL** =(16)
- **UNSUPPORT_FORMAT** =(17)
- **FILEREADER_FAILED** =(18)
- **PLAYSTART_FAILED** =(19)
- **PLAYSTOP_FAILED** =(20)
- **PROJECT_NOT_CREATE** =(21)
- **PROJECT_NOT_OPEN** =(22)
- **CODEC_INIT** =(23)
- **RENDERER_INIT** =(24)
- **THEMESET_CREATE_FAIL** =(25)
- **ADD_CLIP_FAIL** =(26)
- **ENCODE_VIDEO_FAIL** =(27)

- **INPROGRESS_GETCLIPINFO** =(28)
- **THUMBNAIL_BUSY** =(29)
- **UNSUPPORT_MIN_DURATION** =(30)
- **UNSUPPORT_MAX_RESOLUTION** =(31)
- **UNSUPPORT_MIN_RESOLUTION** =(32)
- **UNSUPPORT_VIDEIO_PROFILE** =(33)
- **UNSUPPORT_VIDEO_LEVEL** =(34)
- **UNSUPPORT_VIDEO_FPS** =(35)
- **TRANSCODING_BUSY** =(36)
- **TRANSCODING_NOT_SUPPORTED_FORMAT** =(37)
- **TRANSCODING_USER_CANCEL** =(38)
- **TRANSCODING_NOT_ENOUGHT_DISK_SPACE** =(39)
- **TRANSCODING_CODEC_FAILED** =(40)
- **EXPORT_WRITER_INVAILED_HANDLE** =(41)
- **EXPORT_WRITER_INIT_FAIL** =(42)
- **EXPORT_WRITER_START_FAIL** =(43)
- **EXPORT_AUDIO_DEC_INIT_FAIL** =(44)
- **EXPORT_VIDEO_DEC_INIT_FAIL** =(45)
- **EXPORT_VIDEO_ENC_FAIL** =(46)
- **EXPORT_VIDEO_RENDER_INIT_FAIL** =(47)
- **EXPORT_NOT_ENOUGHT_DISK_SPACE** =(48)
- **UNSUPPORT_AUDIO_PROFILE** =(49)
- **THUMBNAIL_INIT_FAIL** =(50)
- **UNSUPPORT_AUDIO_CODEC** =(51)
- **UNSUPPORT_VIDEO_CODEC** =(52)
- **HIGHLIGHT_FILEREADER_INIT_ERROR** =(53)
- **HIGHLIGHT_TOO_SHORT_CONTENTS** =(54)
- **HIGHLIGHT_CODEC_INIT_ERROR** =(55)
- **HIGHLIGHT_CODEC_DECODE_ERROR** =(56)
- **HIGHLIGHT_RENDER_INIT_ERROR** =(57)
- **HIGHLIGHT_WRITER_INIT_ERROR** =(58)
- **HIGHLIGHT_WRITER_WRITE_ERROR** =(59)
- **HIGHLIGHT_GET_INDEX_ERROR** =(60)
- **HIGHLIGHT_USER_CANCEL** =(61)
- **GETCLIPINFO_USER_CANCEL** =(62)
- **DIRECTEXPORT_CLIPLIST_ERROR** =(63)
- **DIRECTEXPORT_CHECK_ERROR** =(64)
- **DIRECTEXPORT_FILEREADER_INIT_ERROR** =(65)
- **DIRECTEXPORT_FILEWRITER_INIT_ERROR** =(66)
- **DIRECTEXPORT_DEC_INIT_ERROR** =(67)
- **DIRECTEXPORT_DEC_INIT_SURFACE_ERROR** =(68)
- **DIRECTEXPORT_DEC_DECODE_ERROR** =(69)
- **DIRECTEXPORT_ENC_INIT_ERROR** =(70)
- **DIRECTEXPORT_ENC_ENCODE_ERROR** =(71)
- **DIRECTEXPORT_ENC_INPUT_SURFACE_ERROR** =(72)
- **DIRECTEXPORT_ENC_FUNCTION_ERROR** =(73)
- **DIRECTEXPORT_ENC_DSI_DIFF_ERROR** =(74)
- **DIRECTEXPORT_ENC_FRAME_CONVERT_ERROR** =(75)
- **DIRECTEXPORT_RENDER_INIT_ERROR** =(76)
- **DIRECTEXPORT_WRITER_WRITE_ERROR** =(77)
- **DIRECTEXPORT_WRITER_UNKNOWN_ERROR** =(78)
- **FASTPREVIEW_USER_CANCEL** =(79)
- **FASTPREVIEW_CLIPLIST_ERROR** =(80)
- **FASTPREVIEW_FIND_CLIP_ERROR** =(81)
- **FASTPREVIEW_FIND_READER_ERROR** =(82)
- **FASTPREVIEW_VIDEO_RENDERER_ERROR** =(83)
- **FASTPREVIEW_DEC_INIT_SURFACE_ERROR** =(84)
- **HW_NOT_ENOUGH_MEMORY** =(85)
- **EXPORT_USER_CANCEL** =(86)

- **FASTPREVIEW_DEC_INIT_ERROR** =(87)
- **FASTPREVIEW_FILEREADER_INIT_ERROR** =(88)
- **FASTPREVIEW_TIME_ERROR** =(89)
- **FASTPREVIEW_RENDER_INIT_ERROR** =(90)
- **FASTPREVIEW_OUTPUTSURFACE_INIT_ERROR** =(91)
- **FASTPREVIEW_BUSY** =(92)
- **CODEC_DECODE** =(93)
- **RENDERER_AUDIO** =(94)

4.42.1 Detailed Description

This enumeration defines the possible error values of the `nexEngine`.

Since

version 1.3.43

4.43 com.nexstreaming.nexeditorsdk.nexFont Class Reference

Public Member Functions

- String getId ()
- String getSampleText ()
- boolean isBuiltinFont ()
- boolean isSystemFont ()
- Bitmap getSampleImage (Context appContext)
- Typeface getTypeFace ()

Static Public Member Functions

- static List< nexFont > getPresetList ()
- static String[] getFontIds ()
- static Typeface getTypeface (Context appContext, String fontId)
- static void reload ()

4.43.1 Detailed Description

This class allows to bring a built-in font of the NexEditor™SDK as a `Typeface` instance.

Example code :

```
TextPaint p = cachedTextPaint;
 p.setTextSize(textSize);
 p.setAntiAlias(true);
 p.setColor(0xFFFFFFFF);
 Typeface userFont = null;

 String[] fontIds = nexFont.getFontIds();
 if( fontIds.length > 0 ) {
     userFont = nexFont.getTypeface(mContext,fontIds[0]);
 }
```

```
if( userFont!=null ) {
    p.setTypeface(userFont);
}
```

4.43.2 Member Function Documentation

4.43.2.1 static String [] com.nexstreaming.nexeditorsdk.nexFont.getFontIds ( ) [static]

This method gets the IDs of built-in fonts of NexEditor™SDK.

The ID is for internal use only, therefore it may differ with the Android font names.

Returns

The list of built-in font IDs.

Since

version 1.5.23

4.43.2.2 String com.nexstreaming.nexeditorsdk.nexFont.getId ( )

This gets the ID of a font.

Returns

The ID of a font.

Since

1.7.0

4.43.2.3 static List<nexFont> com.nexstreaming.nexeditorsdk.nexFont.getPresetList ( ) [static]

This gets the list of all built-in and downloaded fonts from the Asset Store to nexFont.

Returns

The font list.

Since

1.7.0

4.43.2.4 Bitmap com.nexstreaming.nexeditorsdk.nexFont.getSampleImage (Context *appContext*)

This gets the thumbnail image of a font. A 1000 X 100 pixel size image.

Parameters

appContext	getApplicationContext()

Returns

A 1000 x 100 size font sample image.

Since

1.7.0

4.43.2.5 String com.nexstreaming.nexeditorsdk.nexFont.getSampleText ( )

This is the text for the sample image of a font.

Returns

The text for the sample image of a font.

See Also

getSampleImage(Context)

Since

1.7.0

4.43.2.6 Typeface com.nexstreaming.nexeditorsdk.nexFont.getTypeFace ( )

This gets the Typeface of a font.

Returns

The Typeface.

Since

1.7.0

4.43.2.7 static Typeface com.nexstreaming.nexeditorsdk.nexFont.getTypeface (Context *appContext,* String *fontId*) [static]

This method gets a built-in SDK font as a `Typeface` instance. Also, this method can change all graphic fonts on Android to `Typeface`.

Parameters

appContext	The context that has the built-in SDK font.
fontId	The ID of built-in SDK font which can be retrieved by `getFontIds()`.

Returns

The font as in `Typeface`.

Since

version 1.5.23

4.43.2.8 boolean com.nexstreaming.nexeditorsdk.nexFont.isBuiltinFont ( )

This notifies whether or not the font is built-in the application.

Returns

`TRUE` - built-in font , `FALSE` - Asset Store font

Since

1.7.0

4.43.2.9 boolean com.nexstreaming.nexeditorsdk.nexFont.isSystemFont ( )

This notifies whether or not it is an Android system font.

Returns

true - system font , false - user font

Since

1.7.0

4.43.2.10 static void com.nexstreaming.nexeditorsdk.nexFont.reload ( ) `[static]`

This method updates the Fonts. If new fonts are downloaded from the Asset Store or an existing font was deleted, the developer should use this method to update the internal list.

See Also

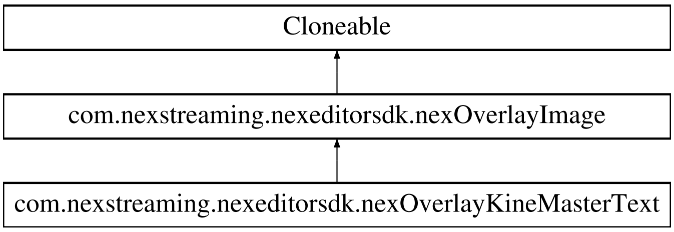
getPresetList()

Since

1.7.0

4.44 com.nexstreaming.nexeditorsdk.nexOverlayImage Class Reference

Inheritance diagram for com.nexstreaming.nexeditorsdk.nexOverlayImage:


Cloneable
com.nexstreaming.nexeditorsdk.nexOverlayImage
com.nexstreaming.nexeditorsdk.nexOverlayKineMasterText

Classes

- interface runTimeMakeBitMap
- class VideoClipInfo

Public Member Functions

- String getId ()
- nexOverlayImage (String itemId, String bitmapPath)
- nexOverlayImage (String itemId, Context context, int resourceId)
- nexOverlayImage (String itemId, runTimeMakeBitMap makeBitmap)
- nexOverlayImage (String itemId, nexClip clip)
- int getResourceId ()
- int getWidth ()
- int getHeight ()
- void setCrop (int width, int height)
- void releaseBitmap ()
- VideoClipInfo getVideoClipInfo ()

Protected Member Functions

- Bitmap **getUserBitmap** ()
- int **getUserBitmapID** ()

Static Protected Member Functions

- static nexOverlayImage **clone** (nexOverlayImage src)

Protected Attributes

- final int **mResourceId**
- boolean **mUpdate** = false

4.44.1 Detailed Description

This class is used to define and manage overlay images in the NexEditor™ SDK.

In the context of the NexEditor™ SDK, an overlay image is any image that will be overlaid onto video or images in a NexEditor™ project, such as for example stickers. This class can be used to add new overlay images to be used as stickers within the NexEditor™ SDK.

To use an overlay image:

1. The overlay image item must first be created with either `nexOverlayImage(String itemId, Bitmap bitmap)` or `nexOverlayImage(String itemId, int resourceId)` for images that will be used more than once.
2. The created overlay image instance must be registered by calling `registerOverlayImage()`.
3. To use an overlay item, it can be called with the overlay image ID set when it is registered.
4. When finished, the overlay image should be unregistered by calling `unregisterOverlayImage()`.

Example code :

```
nexOverlayImage overlay = new nexOverlayImage( "testOverlay",R.drawable.myavatar_17004763_1);
nexOverlayImage.registerOverlayImage(overlay);
```

See Also

nexOverlayItem

Since

version 1.1.0

4.44.2 Constructor & Destructor Documentation

4.44.2.1 com.nexstreaming.nexeditorsdk.nexOverlayImage.nexOverlayImage (String *itemId,* String *bitmapPath*)

This method creates an instance of an overlay image from a bitmap path.

Because this method of creating an overlay image requires a lot of memory, this method should be used to create overlay images that will only be used once.

Example code :

```
 nexOverlayImage makeKmOverlayImage(String imageId, String bitmapPath, int width, int height)
{
nexOverlayImage image = new nexOverlayImage(imageId, bitmapPath);
image.setCrop(width, height);
return image;
}
```

Parameters

itemId	
bitmapPath	

Exceptions

NotSupportedAPILevel	nexEditorSDK API Limited Level is { EditorGlobal::kOverlayImageLimited}

See Also

nexOverlayImage(String itemId, String bitmapPath)

Since

version 1.1.0

4.44.2.2 com.nexstreaming.nexeditorsdk.nexOverlayImage.nexOverlayImage (String *itemId,* Context *context,* int *resourceId*)

This method creates an overlay image from an Android image resource ID.

This method should be used to create overlay images such as stickers and other images created from the registered presets.

Example code :

```
nexOverlayImage overlay = new nexOverlayImage( "testOverlay",this,R.drawable.myavatar_17004763_1);
```

Parameters

itemId	The designated ID to use for the instance of the overlay image item to be created, as a `String`.
context	The Android Application Context.
resourceId	The Android image resource ID of the overlay image to be created, as an `integer`.

Exceptions

NotSupportedAPILevel	nexEditorSDK API Limited Level is { EditorGlobal::kOverlayImageLimited}

See Also

nexOverlayImage(String, Context, int)

Since

version 1.1.0

4.44.2.3 com.nexstreaming.nexeditorsdk.nexOverlayImage.nexOverlayImage (String *itemId,* runTimeMakeBitMap *makeBitmap*)

This method creates an overlay image from the bitmap added by the developer, using the interface `runTimeMakeBitMap`. This method should be used to create overlay images such as stickers and other images created from the registered presets.

Parameters

itemId	The designated ID to use for the instance of the overlay image item to be created, as a `String`.
makeBitmap	The bitmap image made by the developer with `runTimeMakeBitMap`.

Example code :

```
public class TextOverlay implements nexOverlayImage.runTimeMakeBitMap {
                 private String mText;
                 private Context mContext;
                 public TextOverlay(Context context, String text){
                 mText = text;
                 mContext = context;
                 }

                 public boolean isAniMate() {
                 return false;
                 }

                 public int getBitmapID() {
                 return 2;
                 }

                 public Bitmap makeBitmap() {
                 myView vm = new myView(mContext);

                 Bitmap bitmap = Bitmap.createBitmap(200, 200, Bitmap.Config.ARGB_8888);
                 Canvas c = new Canvas(bitmap);
                 vm.draw(c);
                 return bitmap;
                 }
                 ......
                 }

                 nexOverlayImage textOverlay = new nexOverlayImage( "text", new TextOverlay(this,"
      Nexstreaming"));
```

Exceptions

NotSupportedAPILevel	nexEditorSDK API Limited Level is { EditorGlobal::kOverlayImageLimited}

Since

version 1.3.0

4.44.2.4 com.nexstreaming.nexeditorsdk.nexOverlayImage.nexOverlayImage (String *itemId,* nexClip *clip*)

Constructor of the `nexOverlayImage`.

Example code :

```
 nexClip clip = nexClip.getSupportedClip(filepath);
id = mSelectedPIP.size();
nexOverlayImage ov = new nexOverlayImage("video"+id,clip);
```

Parameters

itemId	The designated ID of an overlay image instance to be created, as a `String`.
clip	The media content to be added as an overlay image.

Exceptions

ClipIsNotVideoException	
NotSupportedAPILevel	nexEditorSDK API Limited Level is { EditorGlobal::kOverlayVideoLimited}

Since

version 1.3.52

4.44.3 Member Function Documentation

4.44.3.1 int com.nexstreaming.nexeditorsdk.nexOverlayImage.getHeight ( )

This method gets the height of a clip.

Returns

clip height

See Also

getWidth()

Since

version 1.5.15

4.44.3.2 String com.nexstreaming.nexeditorsdk.nexOverlayImage.getId ( )

This method gets the ID of an overlay image item, as a `String`.

The ID of an overlay image item can be used to retrieve the image, or to unregister the overlay image item when finished with it.

Example code :

```
 nexOverlayImage overlay = new nexOverlayImage( "testOverlay",this,R.drawable.ilove0001);
String overlayid = overlay.getId();
```

Returns

The ID of the overlay image item, set when the item was created, as a `String`.

See Also

#getOverlayImage
#unregisterOverlayImage

Since

version 1.1.0

4.44.3.3 int com.nexstreaming.nexeditorsdk.nexOverlayImage.getResourceId ( )

This method gets the resource ID of an overlay image, if the overlay image was made with a resource ID.

Example code :

```
 nexOverlayImage overlay = new nexOverlayImage("testOverlay",DirectExportTestActivity.this, R.drawable.
      ilove0001);
int rID = overlay.getResourceId();
```

Returns

The resource ID if the overlay image was made with nexOverlayImage(String, Context, int); otherwise 0.

Since

version 1.3.0

4.44.3.4 VideoClipInfo com.nexstreaming.nexeditorsdk.nexOverlayImage.getVideoClipInfo ( )

This method gets information of a video clip. The information includes details on the video content such as width, height, size and etc.

Example code :

```
visualClip.mTotalTime = item.getOverlayImage().getVideoClipInfo().getTotalTime();
```

Returns

VideoClipInfo

Since

version 1.5.0

4.44.3.5 int com.nexstreaming.nexeditorsdk.nexOverlayImage.getWidth ( )

This method gets the width of a clip.

Returns

clip width

See Also

getHeight()

Since

version 1.5.15

4.44.3.6 void com.nexstreaming.nexeditorsdk.nexOverlayImage.releaseBitmap ( )

This method clears the cache, where the bitmap images made with nexOverlayImage(String, com.nexstreaming.nexeditorsdk.nexOverlayImage.runTimeMakeBitMap) are saved, to free up storage space.

Example code :

```
 nexOverlayImage image = new nexOverlayImage(imageId, bitmapPath);
...
image.releaseBitmap();
```

Since

version 1.3.0

4.44.3.7 void com.nexstreaming.nexeditorsdk.nexOverlayImage.setCrop (int *width,* int *height*)

This method crops to resize an overlay image.

Example code :

```
 nexOverlayImage image = new nexOverlayImage(imageId, bitmapPath);
image.setCrop(width, height);
```

Parameters

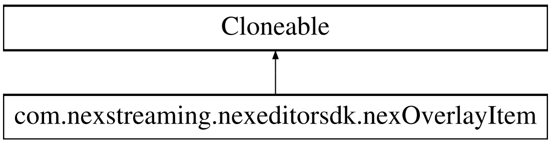
width	The width of the overlay image in `bitmap` as an `integer`.
height	The height of the overlay image in `bitmap` as an `integer`.

Since

version 1.3.38

4.45 com.nexstreaming.nexeditorsdk.nexOverlayItem Class Reference

Inheritance diagram for com.nexstreaming.nexeditorsdk.nexOverlayItem:



Classes

- class BoundInfo
- class ChromaKey
- class HitPoint
- class Mask

Public Member Functions

- void setLayerExpression (nexOverlayKineMasterExpression layerExpressionIndex)
- nexOverlayKineMasterExpression getLayerExpression ()
- void setLayerExpressionDuration (int duration)
- int getLayerExpressionDuration ()
- void setColorEffect (nexColorEffect effect)
- int getId ()
- nexOverlayItem (String overlayImageId, int posX, int posY, int startTime, int endTime)
- nexOverlayItem (nexOverlayImage olImage, int posX, int posY, int startTime, int endTime)
- **nexOverlayItem** (nexOverlayKineMasterText kmText, int posX, int posY, int startTime, int endTime)
- **nexOverlayItem** (nexOverlayFilter filter, int posX, int posY, int startTime, int endTime)
- boolean getAudioOnOff ()
- void setAudioOnOff (boolean audioOnOff)
- int getVolume ()
- void setVolume (int volume)
- void setSpeedControl (int speed)
- int getSpeedControl ()
- int getPositionX ()
- int getPositionY ()
- void setPosition (int x, int y)
- float getAlpha ()
- void setAlpha (float value)
- float getScaledX ()
- float getScaledY ()
- float getScaledZ ()
- void setScale (float scaledX, float scaledY)
- void setScale (float scaledX, float scaledY, float scaledZ)
- int getRotate ()
- int getRotateX ()
- int getRotateY ()
- int getRotateZ ()
- void setTrim (int startTime, int endTime)
- void clearTrim ()
- int getStartTrimTime ()
- int getEndTrimTime ()
- void setRotate (int degree)
- void setRotate (int degreeX, int degreeY, int degreeZ)
- int getBrightness ()
- int getContrast ()
- int getSaturation ()
- boolean setBrightness (int value)
- boolean setContrast (int value)
- boolean setSaturation (int value)
- void setLayerExpressionParam (boolean bApplayLayerExpression)
- boolean getLayerExpressiontParam ()
- int getStartTime ()
- int getEndTime ()
- void setTimePosition (int startTime, int endTime)
- int getAnimateEndTime ()
- void addAnimate (nexAnimate animate)
- void clearAnimate ()
- void clearCache ()
- ChromaKey **getChromaKey** ()
- Mask getMask ()
- BoundInfo getBoundInfo (int time)
- boolean isPointInOverlayItem (HitPoint PointInOut)
- void showOutline (boolean show)

Static Public Member Functions

- static void setOutLine ()
- static void setOutlineType (boolean solid)
- static void clearOutLine ()
- static boolean setOutLineIcon (Context context, int pos, int resID)

Static Public Attributes

- static final int kOutLine_Pos_LeftTop = 1
- static final int kOutLine_Pos_RightTop = 2
- static final int kOutLine_Pos_LeftBottom = 3
- static final int kOutLine_Pos_RightBottom = 4

Static Protected Member Functions

- static nexOverlayItem **clone** (nexOverlayItem src)

Protected Attributes

- int **mShowRangemode**

4.45.1 Detailed Description

This class is used to define and register an overlay image to a project.

This class allows the start time and duration of an overlay image to be set as well as setting possible animation effects. Also, simple color effects and adjustments can be made with this class.

Effects and animations on clips are set in different layers which can be adjusted independently.

The coordinates of an overlay image with ratio of 16:9: upper left (0,0) and bottom right (1280,720). The coordinates of an overlay image with ratio of 1:1: upper left (0,0) and bottom right (720,720).

Example code :

```
 nexOverlayImage overlay = new nexOverlayImage( "testOverlay",R.drawable.myavatar_17004763_1);
nexOverlayImage.registerOverlayImage(overlay);
nexOverlayItem overlayItem = new nexOverlayItem("testOverlay",200,200,0,0+6000)
      ;
nexAnimate ani = nexAnimate.getAnimateImages(
   0,5000,R.drawable.myavatar_17004763_1
   ,R.drawable.myavatar_17004763_2
   ,R.drawable.myavatar_17004763_3
   ,R.drawable.myavatar_17004763_4
   ,R.drawable.myavatar_17004763_5
   ,R.drawable.myavatar_17004763_6
   ,R.drawable.myavatar_17004763_7
   ,R.drawable.myavatar_17004763_8
   ,R.drawable.myavatar_17004763_9
   ,R.drawable.myavatar_17004763_10
   );
   overlayItem.addAnimate(ani);
   project.addOverlay(overlayItem);
```

Since

version 1.1.0

4.45.2 Constructor & Destructor Documentation

4.45.2.1 com.nexstreaming.nexeditorsdk.nexOverlayItem.nexOverlayItem (String *overlayImageId,* int *posX,* int *posY,* int *startTime,* int *endTime*)

This method creates an overlay item with the ID used to register an image to nexOverlayImage.

The coordinates of an overlay image with ratio of 16:9 : upper left (0,0) and bottom right (1280,720). The coordinates of an overlay image with ratio of 1:1 : upper left (0,0) and bottom right (720,720).

Parameters

overlayImageId	The ID used to register an image to nexOverlayImage.
posX	The x-coordinate of the overlay image's center.
posY	The y-coordinate of the overlay image's center.
startTime	The start time of the overlay item (based on the total playtime of the project) in msec (milliseconds).
endTime	The end time of the overlay item (based on the total playtime of the project) in msec (milliseconds).

Example code :

```
 nexOverlayImage overlay2 = new nexOverlayImage( "testOverlay2",R.drawable.ilove0001);
nexOverlayImage.registerOverlayImage(overlay2);
    nexOverlayItem overlayItem2 = new nexOverlayItem("testOverlay2",800,200,0,0
      +6000);
```

Since

version 1.1.0

4.45.2.2 com.nexstreaming.nexeditorsdk.nexOverlayItem.nexOverlayItem (nexOverlayImage *oImage,* int *posX,* int *posY,* int *startTime,* int *endTime*)

This method creates an overlay item, that uses an overlay image made by the developer, which can be later added to the project.

The coordinates of an overlay image with ratio of 16:9 : upper left (0,0) and bottom right (1280,720). The coordinates of an overlay image with ratio of 1:1 : upper left (0,0) and bottom right (720,720).

Parameters

oImage	An nexOverlayImage instance of the overlay image.
posX	The x-coordinate of the overlay image's center.
posY	The y-coordinate of the overlay image's center.
startTime	The start time of the overlay item (based on the total playtime of the project) in msec (milliseconds).

endTime	The end time of the overlay item (based on the total playtime of the project) in `msec` (milliseconds).
	`nexOverlayImage textOverlay = new nexOverlayImage( "text",new TextOverlay(this,"Nexstreaming"));`
	`nexOverlayItem overlayItem3 = new nexOverlayItem(textOverlay,200,400,0,6000);`

Since

version 1.1.0

<font color="#ffffff">nexEditorSDK API Limited Level</font>	{ EditorGlobal::kOverlayImageLimited}

4.45.3 Member Function Documentation

4.45.3.1 void com.nexstreaming.nexeditorsdk.nexOverlayItem.addAnimate (nexAnimate *animate*)

This method adds any animation effect customized by the developer to the class `nexOverlayItem`.

Example code :

```
nexAnimate ani = nexAnimate.getAnimateImages(
    0, 5000, R.drawable.ilove0001
    , R.drawable.ilove0002
    , R.drawable.ilove0003
    , R.drawable.ilove0004
    , R.drawable.ilove0005
    , R.drawable.ilove0006
    , R.drawable.ilove0007
    , R.drawable.ilove0008
    , R.drawable.ilove0009
    , R.drawable.ilove0010
    );
overlayItem.addAnimate(ani);
```

Parameters

animate	An instance of the animation effect to add to the NexEditor™ SDK.

See Also

clearAnimate()

Since

version 1.1.0

4.45.3.2 void com.nexstreaming.nexeditorsdk.nexOverlayItem.clearAnimate ( )

This method clears all the customized animation effects added by a developer.

Example code :

```
overlayItem.clearAnimate();
```

See Also

addAnimate(nexAnimate)

Since

version 1.1.0

4.45.3.3 void com.nexstreaming.nexeditorsdk.nexOverlayItem.clearCache ( )

This method clears the overlay item cache.

Example code :

```
for( nexOverlayItem temp : mProjectOverlays ){
    Log.d(TAG,"Overlay clear id = "+temp.getId());
    temp.clearCache();
}
mProjectOverlays.clear();
```

Since

version 1.3.23

4.45.3.4 static void com.nexstreaming.nexeditorsdk.nexOverlayItem.clearOutLine ( ) [static]

This method clears the dashed outline resources for when the overlay edit is finished.

See Also

setOutLine()

Since

version 1.5.3

4.45.3.5 void com.nexstreaming.nexeditorsdk.nexOverlayItem.clearTrim ( )

This method clears the start time and end time set to the method `setTrim`.

Example code :

```
if( mEndTrimTime <=  mStartTrimTime ){
    mProject.getOverlay(selectedPIPId).clearTrim();
}else {
    mProject.getOverlay(selectedPIPId).setTrim(mStartTrimTime, mEndTrimTime);
}
```

See Also

setTrim(int, int)

Since

version 1.5.0

4.45.3.6 float com.nexstreaming.nexeditorsdk.nexOverlayItem.getAlpha ( )

This method gets the alpha value or transparency of an overlay image.

Example code :

```
nexOverlayItem item = mProject.getOverlay(mSelectedPIP.get(position).
      getId());
float alpha = item.getAlpha();
```

Returns

The value of the alpha set for the overlay image.

See Also

setAlpha(float)

Since

version 1.1.0

4.45.3.7 int com.nexstreaming.nexeditorsdk.nexOverlayItem.getAnimateEndTime ( )

This method gets the end time of an animation effect customized by the developer.

Example code :

```
int endTime = overlayItem.getAnimateEndTime();
```

Returns

The end time of the developer customized animation effect.

Since

version 1.1.0

4.45.3.8 boolean com.nexstreaming.nexeditorsdk.nexOverlayItem.getAudioOnOff ( )

This method gets the audio volume state of a clip, whether it's `on` or `off`. Default value is `on`.

Example code :

```
for( nexOverlayItem item : mProjectOverlays ) {
    visualClip.mAudioOnOff = item.getAudioOnOff()?1:0;
    ...
}
```

Returns

The audio volume state. `TRUE` if the audio state is `on`; otherwise `FALSE`.

See Also

setAudioOnOff(boolean)

Since

version 1.5.0

4.45.3.9 BoundInfo com.nexstreaming.nexeditorsdk.nexOverlayItem.getBoundInfo (int *time*)

This method gets the BoundInfo from the timeline of this overlay item.

Parameters

time	The time of the overlay timeline.

Returns

The BoundInfo that includes the current coordinates.

Since

version 1.5.3

4.45.3.10 int com.nexstreaming.nexeditorsdk.nexOverlayItem.getBrightness ( )

This method gets the brightness of an overlay image.

Example code :

```
final nexOverlayItem overlayItem = new nexOverlayItem(overlay,640,360,0,0+6000)
      ;
int Brightness = overlayItem.getBrightness();
```

Returns

The brightness of the overlay image, as an `integer` .

See Also

setBrightness(int)

Since

version 1.1.0

4.45.3.11 int com.nexstreaming.nexeditorsdk.nexOverlayItem.getContrast ( )

This method gets the contrast of an overlay image.

Example code :

```
final nexOverlayItem overlayItem = new nexOverlayItem(overlay,640,360,0,0+6000)
      ;
int Contrast = overlayItem.getContrast();
```

Returns

The contrast of the overlay image, as an `integer` between -255 and 255.

See Also

setContrast(int)

Since

version 1.1.0

4.45.3.12 int com.nexstreaming.nexeditorsdk.nexOverlayItem.getEndTime ( )

This method gets the end time of an overlay image (based on the total playtime of the project).

Example code :

```
for( nexOverlayItem temp : mProjectOverlays ){
    if (temp.getStartTime() <= point.mTime && temp.getEndTime() > point.mTime) {
        return temp.isPointInOverlayItem(point);
    }
}
```

Returns

The end time of the overlay image in `msec` (milliseconds), as an `integer`.

See Also

setTimePosition(int, int)

Since

version 1.1.0

4.45.3.13 int com.nexstreaming.nexeditorsdk.nexOverlayItem.getEndTrimTime ( )

This method gets the end time of the section to be trimmed from an overlay item.

Example code :

```
int startTrim = mProject.getOverlay(selectedPIPId).getStartTrimTime();
int endTrim = mProject.getOverlay(selectedPIPId).getEndTrimTime();
```

Returns

The end time of the trimmed section.

See Also

setTrim(int, int)
getStartTrimTime()

Since

version 1.5.0

4.45.3.14 int com.nexstreaming.nexeditorsdk.nexOverlayItem.getId ( )

This method gets the ID of an `nexOverlayItem` instance.

The unique ID of an overlay item is made when a project is created.

Example code :

```
int overlay_id = overlayItem.getId();
```

Returns

The overlay item ID provided when a project is created, as an `integer`.

Since

version 1.1.0

4.45.3.15 nexOverlayKineMasterExpression com.nexstreaming.nexeditorsdk.nexOverlayItem.getLayerExpression ( )

Deprecated This method gets the index of a preset animation effect. Use `getLayerExpressions()` to get the label names.

Example code :

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>(getApplicationContext(),
R.layout.simple_spinner_dropdown_item_1, nexOverlayItem.getLayerExpressions());
```

Returns

The index of the preset animation effect, as an `integer`.

Since

version 1.1.0

See Also

setLayerExpression(nexOverlayKineMasterExpression)

Deprecated For internal use only. Please do not use.

4.45.3.16 int com.nexstreaming.nexeditorsdk.nexOverlayItem.getLayerExpressionDuration ( )

Deprecated This method gets the duration of a preset animation effect for a clip.

Example code :

```
int overlay_duration = overlayItem.getLayerExpressionDuration();
```

Returns

The duration of the preset animation in `msec` (milliseconds), as an `integer`.

See Also

setLayerExpressionDuration(int)

Since

version 1.1.0

4.45.3.17 boolean com.nexstreaming.nexeditorsdk.nexOverlayItem.getLayerExpressiontParam ( )

Returns

Deprecated

4.45.3.18 Mask com.nexstreaming.nexeditorsdk.nexOverlayItem.getMask ( )

This method gets the mask value where an overlay item will be printed.

Example code :

```
mProject.getOverlay(selectedPIPId).getMask().setState(false);
```

Returns

The mask

Since

version 1.5.0

4.45.3.19 int com.nexstreaming.nexeditorsdk.nexOverlayItem.getPositionX ( )

This method gets the x-coordinate of an overlay image's location.

Example code :

```
mEngine.getProject().getOverlay(mOverlayId).setPosition(mEngine.getProject().getOverlay(mOverlayId)
      .getPositionX(), lastProgress);
```

Returns

The value of x-coordinate of the center of the overlay image.

See Also

setPosition(int, int)

Since

version 1.1.0

4.45.3.20 int com.nexstreaming.nexeditorsdk.nexOverlayItem.getPositionY ( )

This method gets the y-coordinate of an overlay image's location.

Example code :

```
mEngine.getProject().getOverlay(mOverlayId).setPosition(lastProgress, mEngine.getProject().getOverlay(
      mOverlayId).getPositionY());
```

Returns

The value of y-coordinate of the center of the overlay image.

See Also

setPosition(int, int)

Since

version 1.1.0

4.45.3.21 int com.nexstreaming.nexeditorsdk.nexOverlayItem.getRotate ( )

This method gets the rotation value of an overlay image.

Example code :

```
nexOverlayItem item = mProject.getOverlay(mSelectedPIP.get(position).
      getId());
int rotate = item.getRotate();
```

Returns

The rotation value of an overlay image as an `integer`.

See Also

setRotate(int)

Since

version 1.1.0

4.45.3.22 int com.nexstreaming.nexeditorsdk.nexOverlayItem.getRotateX ( )

This method gets the x-axis rotation value of an overlay image.

Example code :

```
private nexOverlayItem mItem;
mItem.setRotate(mItem.getRotateX(), degree,mItem.getRotateZ());
```

Returns

The x-axis rotation value.

See Also

getRotateY()
getRotateZ()

Since

version 1.3.42

4.45.3.23 int com.nexstreaming.nexeditorsdk.nexOverlayItem.getRotateY ( )

This method gets the y-axis rotation value of an overlay image.

Example code :

```
private nexOverlayItem mItem;
mItem.setRotate(mItem.getRotateY(), degree,mItem.getRotateX());
```

Returns

The y-axis rotation value.

See Also

getRotateX()
getRotateZ()

Since

version 1.3.42

4.45.3.24 int com.nexstreaming.nexeditorsdk.nexOverlayItem.getRotateZ ( )

This method gets the z-axis rotation value of an overlay image.

Example code :

```
private nexOverlayItem mItem;
mItem.setRotate(mItem.getRotateX(), degree,mItem.getRotateZ());
```

Returns

The z-axis rotation value.

See Also

getRotateY()
getRotateX()

Since

version 1.3.42

4.45.3.25 int com.nexstreaming.nexeditorsdk.nexOverlayItem.getSaturation ( )

This method gets the saturation of an overlay image.

Example code :

```
final nexOverlayItem overlayItem = new nexOverlayItem(overlay,640,360,0,0+6000)
      ;
int Saturation = overlayItem.getSaturation();
```

Returns

The saturation of the overlay image, as an `integer` between -255 and 255.

See Also

setSaturation(int)

Since

version 1.1.0

4.45.3.26 float com.nexstreaming.nexeditorsdk.nexOverlayItem.getScaledX ( )

This method gets the scale factor for width of an overlay image.

Example code :

```
private nexOverlayItem mItem;
mItem.setScale(mItem.getScaledX(),ratio);
```

Returns

The scale factor for the width of the overlay image.

See Also

setScale(float, float)

Since

version 1.1.0

4.45.3.27 float com.nexstreaming.nexeditorsdk.nexOverlayItem.getScaledY ( )

This method gets the scale factor for height of an overlay image.

Example code :

```
private nexOverlayItem mItem;
mItem.setScale(mItem.getScaledY(),ratio);
```

Returns

The scale factory for the height of the overlay image.

See Also

setScale(float, float)

Since

version 1.1.0

4.45.3.28 float com.nexstreaming.nexeditorsdk.nexOverlayItem.getScaledZ ( )

Returns

Since

version 1.5.1

4.45.3.29 int com.nexstreaming.nexeditorsdk.nexOverlayItem.getSpeedControl ( )

This method gets the playback speed (in range of 13 ∼ 400) of a clip in an `integer` Default is value 100.

Example code :

```
for( nexOverlayItem item : mProjectOverlays ) {
    visualClip.mClipSpeed = item.getSpeedControl();
    ...
}
```

Returns

The playback speed of the clip as an `integer`.

See Also

setSpeedControl(int)

Since

version 1.5.0

4.45.3.30 int com.nexstreaming.nexeditorsdk.nexOverlayItem.getStartTime ( )

This method gets the start time of an overlay image (based on the total playtime of the project).

Example code :

```
for( nexOverlayItem temp : mProjectOverlays ){
    if (temp.getStartTime() <= point.mTime && temp.getEndTime() > point.mTime) {
        return temp.isPointInOverlayItem(point);
    }
}
```

Returns

The start time of the overlay image.

See Also

setTimePosition(int, int)

Since

version 1.1.0

4.45.3.31 int com.nexstreaming.nexeditorsdk.nexOverlayItem.getStartTrimTime ( )

This method gets the start time of the section to be trimmed from an overlay item.

Example code :

```
int startTrim = mProject.getOverlay(selectedPIPId).getStartTrimTime();
int endTrim = mProject.getOverlay(selectedPIPId).getEndTrimTime();
```

Returns

The start time of the trimmed section.

See Also

setTrim(int, int)
getEndTrimTime()

Since

version 1.5.0

4.45.3.32 int com.nexstreaming.nexeditorsdk.nexOverlayItem.getVolume ( )

This method gets the audio volume level (in range of 0 ∼ 200) of a clip as an `integer`. Default value is 100.

Example code :

```
for( nexOverlayItem item : mProjectOverlays ) {
    visualClip.mClipVolume = item.getVolume();
    ...
}
```

Returns

The volume level of the clip.

See Also

setVolume(int)

Since

version 1.5.0

4.45.3.33 boolean com.nexstreaming.nexeditorsdk.nexOverlayItem.isPointInOverlayItem (HitPoint *PointInOut*)

This API checks whether or not this overlay exists in the entered HitPoint.

Example code :

```
 public boolean onTouch(View v, MotionEvent event) {
    int act = event.getAction();
    if ((act&MotionEvent.ACTION_MASK) == MotionEvent.ACTION_DOWN) {
        mPosition.mTime = 0;
        mPosition.mViewX = event.getX();
        mPosition.mViewY = event.getY();
        mPosition.mViewWidth = v.getWidth();
        mPosition.mViewHeight = v.getHeight();
        item.isPointInOverlayItem(mPosition )
}
}
```

Parameters

PointInOut	The values of HitPoint.mTime, HitPoint.mViewX, HitPoint.mViewY, HitPoint.mViewWidth, HitPoint.mViewHeight must be present for this class to function.

Returns

TRUE if a point is hit; otherwise, FALSE.

Since

version 1.5.3

4.45.3.34 void com.nexstreaming.nexeditorsdk.nexOverlayItem.setAlpha (float *value*)

This method sets the alpha value or transparency of an overlay image.

Example code :

```
mProject.getOverlay(selectedPIPId).setAlpha(mSelectedItemAlpha);
```

Parameters

value	The value of alpha to set to the overlay image, in range of 0 to 1.

See Also

getAlpha()

Since

version 1.1.0

4.45.3.35 void com.nexstreaming.nexeditorsdk.nexOverlayItem.setAudioOnOff (boolean *audioOnOff*)

This method sets the audio volume state of a clip, whether it's `on` or `off`. Default value is `on`.

Example code :

```
for( nexOverlayItem item : mProjectOverlays ) {
    item.setAudioOnOff(false);
    ...
}
```

Parameters

audioOnOff	The audio volume state. `TRUE` if the audio state is `on`, otherwise `FALSE`.

See Also

getAudioOnOff()

Since

version 1.5.0

4.45.3.36 boolean com.nexstreaming.nexeditorsdk.nexOverlayItem.setBrightness (int *value*)

This method sets the brightness for an overlay image.

Example code :

```
final nexOverlayItem overlayItem = new nexOverlayItem(overlay,640,360,0,0+6000)
      ;
overlayItem.setBrightness(lastProgress);
```

Parameters

value	The brightness to set for the overlay image, in the range of -255 to 255.

Returns

`TRUE` when successful, otherwise `FALSE` when the value is not in the permitted range.

See Also

getBrightness()

Since

version 1.1.0

4.45.3.37 void com.nexstreaming.nexeditorsdk.nexOverlayItem.setColorEffect (nexColorEffect *effect*)

This method sets a color effect on an overlay image.

Example code :

```
mEngine.getProject().getClip(0,true).setColorEffect( nexColorEffect.getPresetList().get(position) );
```

Parameters

effect	The color effect, `nexColorEffect`, to set on an overlay image.

Since

version 1.1.0

4.45.3.38 boolean com.nexstreaming.nexeditorsdk.nexOverlayItem.setContrast (int *value*)

This method sets the contrast for an overlay image.

Example code :

```
final nexOverlayItem overlayItem = new nexOverlayItem(overlay,640,360,0,0+6000)
      ;
overlayItem.setContrast(lastProgress);
```

Parameters

value	The contrast to set for the overlay image, in the range of -255 to 255.

Returns

`TRUE` when successful, otherwise `FALSE` when the value is not in the permitted range.

See Also

getContrast()

Since

version 1.1.0

4.45.3.39 void com.nexstreaming.nexeditorsdk.nexOverlayItem.setLayerExpression (nexOverlayKineMasterExpression *layerExpressionIndex*)

Deprecated This method sets an effect for a preset animation. If the parameter `layerExpression` is set to 0, an animation effect made by the developer should be used instead of a preset animation effect.

Example code :

```
mEngin.getProject().getOverlay(mOverlayItemId).setLayerExpression(nexOverlayKineMasterExpression.NONE);
```

Parameters

layerExpressionIndex	The index of the label from the array that is to be retrieved from `getLayerExpressions()`, as `integer`.

Exceptions

com.nexstreaming.nexeditorsdk.exception.InvalidRangeException	

Since

version 1.1.0

See Also

getLayerExpression()

4.45.3.40 void com.nexstreaming.nexeditorsdk.nexOverlayItem.setLayerExpressionDuration (int *duration*)

Deprecated This method sets the duration of a preset animation effect for a clip.

Example code :

```
overlayItem.setLayerExpressionDuration(2000);
```

Parameters

duration	The duration of the preset animation in `msec` (milliseconds), as an `integer`.

See Also

getLayerExpressionDuration()

Since

version 1.1.0

4.45.3.41 void com.nexstreaming.nexeditorsdk.nexOverlayItem.setLayerExpressionParam (boolean *bApplayLayerExpression*)

Parameters

bApplayLayerExpression	

4.45.3.42 static void com.nexstreaming.nexeditorsdk.nexOverlayItem.setOutLine ( ) [static]

This method sets the dashed outline of the selected overlay item.

Use this method to initialize the application when making an application with the overlay edit mode.

See Also

setOutLineIcon(Context, int, int)
setOutlineType(boolean)
clearOutLine()

Since

version 1.5.3

4.45.3.43 static boolean com.nexstreaming.nexeditorsdk.nexOverlayItem.setOutLineIcon (Context *context,* int *pos,* int *resID*) [static]

This method sets the image and position of the icon during the overlay edit. The icon must be included in the resources.

Parameters

context	The Android context where the icon image exists.
pos	The position to where the icon will be set. kOutLine_Pos_LeftTop={ kOutLine_Pos_LeftTop} , kOutLine_Pos_RightTop={ kOutLine_Pos_RightTop} , kOutLine_Pos_LeftBottom={ kOutLine_Pos_LeftBottom} ,kOutLine_Pos_RightBottom={ kOutLine_Pos_RightBottom}
resID	The resource ID of the icon.

Note

In order to show the icon properly, call setOutLine() first.

Returns

TRUE if the icon setting was successful; otherwise, FALSE.

Since

version 1.5.3

4.45.3.44 static void com.nexstreaming.nexeditorsdk.nexOverlayItem.setOutlineType (boolean *solid*) [static]

This method sets the outline type to solid white. The default outline type is dashed black and white.

Parameters

solid	Set to `TRUE` for solid white, or set to `FALSE` for dashed black and white.

Since

version 1.5.3

4.45.3.45 void com.nexstreaming.nexeditorsdk.nexOverlayItem.setPosition (int *x*, int *y*)

This method sets the location of an overlay image.

For overlay images with aspect ratio of 16:9, set the coordinate values based on 1280 X 720 resolution. For overlay images with aspect ratio of 1:1, set the coordinate values based on 720 X 720 resolution.

Example code :

```
mEngine.getProject().getOverlay(mOverlayId).setPosition(lastProgress, mEngine.getProject().getOverlay(
      mOverlayId).getPositionY());
```

Parameters

x	The x-coordinate of the overlay image's center.
y	The y-coordinate of the overlay image's center.

See Also

getPositionX()
getPositionY()

Since

version 1.1.0

4.45.3.46 void com.nexstreaming.nexeditorsdk.nexOverlayItem.setRotate (int *degree*)

This method rotates an overlay image by one of four specific angles (0, 90, 180, 270).

Example code :

```
mEngine.getProject().getOverlay(mOverlayId).setRotate(lastProgress);
```

Parameters

degree	The amount to rotate the overlay image in degrees, as an `integer`. This should be one of four possible angles (0, 90, 180, 270).

See Also

getRotate()

Since

version 1.1.0

4.45.3.47 void com.nexstreaming.nexeditorsdk.nexOverlayItem.setRotate (int *degreeX,* int *degreeY,* int *degreeZ*)

This method gets the rotation value of an overlay image.

Example code :

```
final nexOverlayItem overlayItem = new nexOverlayItem(overlay,640,360,0,0+6000)
      ;
overlayItem.setRotate(lastProgress, overlayItem.getRotateY(), overlayItem.getRotateZ
```

Parameters

degreeX	The x-axis rotation value.
degreeY	The y-axis rotation value.
degreeZ	The z-axis rotation value.

See Also

setRotate(int)

Since

version 1.3.42

4.45.3.48 boolean com.nexstreaming.nexeditorsdk.nexOverlayItem.setSaturation (int *value*)

This method sets the saturation for an overlay image.

Example code :

```
final nexOverlayItem overlayItem = new nexOverlayItem(overlay,640,360,0,0+6000)
      ;
overlayItem.setSaturation(lastProgress);
```

Parameters

value	The saturation value to set for the overlay image, in the range of -255 to 255.

Returns

`TRUE` when successful, otherwise `FALSE` when the value is not in the permitted range.

See Also

getSaturation()

Since

version 1.1.0

4.45.3.49 void com.nexstreaming.nexeditorsdk.nexOverlayItem.setScale (float *scaledX,* float *scaledY*)

This method sets the scale ratio of an overlay image to be scaled.

Example code :

```
private nexOverlayItem mItem;
mItem.setScale(mItem.getScaledX(),ratio);
```

Parameters

scaledX	The scale ratio of the width.
scaledY	The scale ratio of the height.

See Also

getScaledX()
getScaledY()

Since

version 1.1.0

4.45.3.50 void com.nexstreaming.nexeditorsdk.nexOverlayItem.setScale (float *scaledX,* float *scaledY,* float *scaledZ*)

This method sets the scale ratio of an overlay image to be scaled.

Example code :

```
private nexOverlayItem mItem;
mItem.setScale(mItem.getScaledX(),ratio, mItem.getScaledZ());
```

Parameters

scaledX	The scale ratio of the width.
scaledY	The scale ratio of the height.
scaledZ	The scale ratio of the depth.

See Also

getScaledX()
getScaledY()
getScaledZ()

Since

version 1.5.1

4.45.3.51 void com.nexstreaming.nexeditorsdk.nexOverlayItem.setSpeedControl (int *speed*)

This method sets the playback speed (in range of 13 ∼ 400) of a clip. Default value is 100.

Example code :

```
for( nexOverlayItem item : mProjectOverlays ) {
    item.setSpeedControl(200);
    ...
}
```

Parameters

speed	The playback speed of the clip.

See Also

getSpeedControl()

Since

version 1.5.0

4.45.3.52 void com.nexstreaming.nexeditorsdk.nexOverlayItem.setTimePosition (int *startTime,* int *endTime*)

This method sets the duration of an overlay image (based on the total playtime of the project).

The parameter `startTime` sets when the overlay image will be displayed, relative to the project timeline, and `endTime` sets when the overlay image will stop being displayed.

Example code :

```
overlayItem.setTimePosition(startTime, endTime);
```

Parameters

startTime	The start time to set for the overlay image on the project timeline in `msec` (milliseconds).
endTime	The end time to set for the overlay image on the project timeline in `msec` (milliseconds).

See Also

getStartTime()
getEndTime()

Since

version 1.1.0

4.45.3.53 void com.nexstreaming.nexeditorsdk.nexOverlayItem.setTrim (int *startTime,* int *endTime*)

This method sets the section to be trimmed from an overlay item.

Example code :

```
if( mEndTrimTime <=  mStartTrimTime ){
    mProject.getOverlay(selectedPIPId).clearTrim();
} else {
    mProject.getOverlay(selectedPIPId).setTrim(mStartTrimTime, mEndTrimTime);
}
```

Parameters

startTime	The start time of the section to be trimmed.
endTime	The end time of the section to be trimmed.

See Also

clearTrim()

Since

version 1.5.0

4.45.3.54 void com.nexstreaming.nexeditorsdk.nexOverlayItem.setVolume (int *volume*)

This method sets the audio volume level (in range of 0 ∼ 200) of a clip. Default value is 100.

Example code :

```
for( nexOverlayItem item : mProjectOverlays ) {
    item.setVolume(false);
    ...
}
```

Parameters

volume	Clip volume level

See Also

getVolume()

Since

version 1.5.0

4.45.3.55 void com.nexstreaming.nexeditorsdk.nexOverlayItem.showOutline (boolean *show*)

This method shows a dashed outline to indicate that this overlay item has been selected.

Parameters

show	Set to `TRUE` to show an outline, or set to `FALSE` to hide.

Note

setOutLine() must be called first to show a dashed outline properly.

See Also

setOutLine()

Since

version 1.5.3

4.45.4 Member Data Documentation

4.45.4.1 final int com.nexstreaming.nexeditorsdk.nexOverlayItem.kOutLine_Pos_LeftBottom = 3 `[static]`

This method gets the bottom-left corner position of the dashed outline.

Since

version 1.5.3

4.45.4.2 final int com.nexstreaming.nexeditorsdk.nexOverlayItem.kOutLine_Pos_LeftTop = 1 `[static]`

This method gets the top-left corner position of the dashed outline.

Since

version 1.5.3

4.45.4.3 final int com.nexstreaming.nexeditorsdk.nexOverlayItem.kOutLine_Pos_RightBottom = 4 [static]

This method gets the bottom-right corner position of the dashed outline.

Since

version 1.5.3

4.45.4.4 final int com.nexstreaming.nexeditorsdk.nexOverlayItem.kOutLine_Pos_RightTop = 2 [static]

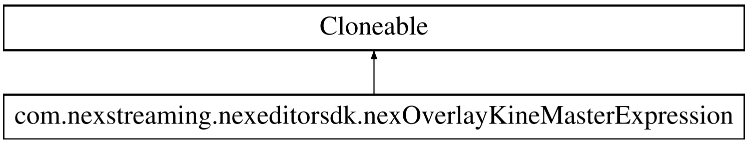
This method gets the top-right corner position of the dashed outline.

Since

version 1.5.3

4.46 com.nexstreaming.nexeditorsdk.nexOverlayKineMasterExpression Class Reference

Inheritance diagram for com.nexstreaming.nexeditorsdk.nexOverlayKineMasterExpression:


Cloneable
com.nexstreaming.nexeditorsdk.nexOverlayKineMasterExpression

Public Member Functions

- int getID ()

Static Public Member Functions

- static List
  < nexOverlayKineMasterExpression > getPresetList ()
- static
  nexOverlayKineMasterExpression[] values ()
- static
  nexOverlayKineMasterExpression getExpression (int id)
- static String[] getNames ()

Static Public Attributes

- static final
  nexOverlayKineMasterExpression **NONE** = new nexOverlayKineMasterExpression("None",0,"none","none","none")
- static final
  nexOverlayKineMasterExpression **FADE** = new nexOverlayKineMasterExpression("Fade",1,"FadeIn","FadeOut","none")

- static final nexOverlayKineMasterExpression **POP** = new nexOverlayKineMasterExpression("Pop",2,"PopIn","ScaleUpOut","none")
- static final nexOverlayKineMasterExpression **SLIDE** = new nexOverlayKineMasterExpression("Slide",3,"SlideRightIn","SlideRightOut","none")
- static final nexOverlayKineMasterExpression **SPIN** = new nexOverlayKineMasterExpression("Spin",4,"SpinCCWIn","SpinCWOut","none")
- static final nexOverlayKineMasterExpression **DROP** = new nexOverlayKineMasterExpression("Drop",5,"DropIn","FadeOut","none")
- static final nexOverlayKineMasterExpression **SCALE** = new nexOverlayKineMasterExpression("Scale",6,"ScaleUpIn","ScaleDownOut","none")
- static final nexOverlayKineMasterExpression **FLOATING** = new nexOverlayKineMasterExpression("Floating",7,"FadeIn","FadeOut","FloatingOverall")
- static final nexOverlayKineMasterExpression **DRIFTING** = new nexOverlayKineMasterExpression("Drifting",8,"FadeIn","FadeOut","DriftingOverall")
- static final nexOverlayKineMasterExpression **SQUISHING** = new nexOverlayKineMasterExpression("Squishing",9,"FadeIn","FadeOut","SquishingOverall")

4.46.1 Detailed Description

This class is a list of predefined animations of the KineMaster™.

The expressions of predefined animations follow the KineMaster™.

Example code :

```
mProject.getOverlay(mOverlayItemId).setLayerExpression(nexOverlayKineMasterExpression.getExpression(
      position));
```

Since

version 1.5.23

4.46.2 Member Function Documentation

4.46.2.1 static nexOverlayKineMasterExpression com.nexstreaming.nexeditorsdk.nexOverlayKineMasterExpression.getExpression (int *id*) [static]

This method gets the ID value of `nexOverlayKineMasterExpression`.

Parameters

id	The ID of the predefined nexOverlayKineMasterExpression which can be found out by `getID()`.

Returns

The animation that corresponds to the parameter `id`.

Exceptions

InvalidRangeException	will occur when the ID does not exist.

Since

version 1.5.23

4.46.2.2 int com.nexstreaming.nexeditorsdk.nexOverlayKineMasterExpression.getID ( )

This method gets the ID value.

Returns

The ID value.

4.46.2.3 static String [ ] com.nexstreaming.nexeditorsdk.nexOverlayKineMasterExpression.getNames ( ) `[static]`

This method gets the names of the animation presets that are compatible with KineMaster™ as an `array`.

Returns

The names of predefined animations.

Since

version 1.5.23

4.46.2.4 static List<nexOverlayKineMasterExpression> com.nexstreaming.nexeditorsdk.nexOverlayKineMasterExpression.getPresetList ( ) `[static]`

This method gets the animation presets that are compatible with KineMaster™ as a list.

Returns

A list of predefined animations.

Since

version 1.5.23

4.46.2.5 static nexOverlayKineMasterExpression [ ] com.nexstreaming.nexeditorsdk.nexOverlayKineMasterExpression.values ( ) [static]

This method gets the animation presets that are compatible with KineMaster™ as an `array`.

Returns

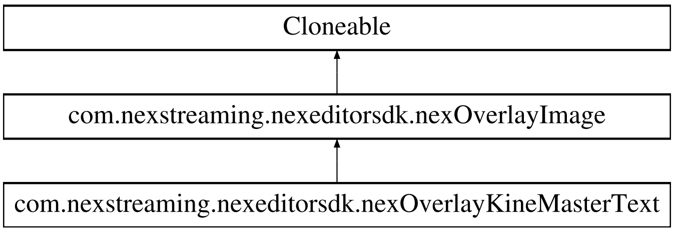
An `array` of predefined animations.

Since

version 1.5.23

4.47 com.nexstreaming.nexeditorsdk.nexOverlayKineMasterText Class Reference

Inheritance diagram for com.nexstreaming.nexeditorsdk.nexOverlayKineMasterText:


Cloneable
com.nexstreaming.nexeditorsdk.nexOverlayImage
com.nexstreaming.nexeditorsdk.nexOverlayKineMasterText

Public Member Functions

- nexOverlayKineMasterText (Context context)
- nexOverlayKineMasterText (Context context, String Text, int textSize)
- int getWidth ()
- int getHeight ()
- String getText ()
- void setText (String text)
- void setTextSize (float size)
- void setTextColor (int argb)
- float getTextSize ()
- int getTextColor ()
- int getGlowColor ()
- void setGlowColor (int argbColor)
- int getOutlineColor ()
- void setOutlineColor (int argbColor)
- int getShadowColor ()
- void setShadowColor (int argbColor)
- boolean isEnableShadow ()
- void EnableShadow (boolean enable)
- boolean isEnableGlow ()
- void EnableGlow (boolean enable)
- boolean isEnableOutline ()
- void EnableOutline (boolean enable)
- void setFontId (String fontId)
- String getFontId ()

Protected Member Functions

- Bitmap **getUserBitmap** ()
- int **getUserBitmapID** ()

Additional Inherited Members

4.47.1 Detailed Description

This is a layer text class which is compatible with the KineMaster™ . Use this class to display an overlay text easily. For compatibility issues with KineMaster™ , there won't be any more expansions.

Example code :

```
nexOverlayKineMasterText builtinText = new
      nexOverlayKineMasterText(getApplicationContext(),"nexOverlayStandardText", 50);
builtinText.setTextColor(0xffff0000);
String[] fontIds = nexFont.getFontIds();
if( fontIds.length > 0 ) {
    builtinText.setFontId(fontIds[0]);
}
nexOverlayItem overlayItem = new nexOverlayItem(builtinText,640,180,0,6000);
project.addOverlay(overlayItem);
```

Since

version 1.5.23

4.47.2 Constructor & Destructor Documentation

4.47.2.1 com.nexstreaming.nexeditorsdk.nexOverlayKineMasterText.nexOverlayKineMasterText (Context *context*)

This is a constructor of nexOverlayKineMasterText.

Parameters

context	The application context with the built-in SDK font.

Since

version 1.5.23

4.47.2.2 com.nexstreaming.nexeditorsdk.nexOverlayKineMasterText.nexOverlayKineMasterText (Context *context,* String *Text,* int *textSize*)

This is a constructor of nexOverlayKineMasterText.

Parameters

context	The application context with the built-in SDK font.
Text	The text to display on the overlay item.
textSize	The size of the text, which follows the size of class `Paint`.

Since

version 1.5.23

4.47.3 Member Function Documentation

4.47.3.1 void com.nexstreaming.nexeditorsdk.nexOverlayKineMasterText.EnableGlow (boolean *enable*)

This method enables the glow effect on the overlay item.

Parameters

enable	`True` if the effect is enabled; otherwise `false`.

Since

version 1.5.23

4.47.3.2 void com.nexstreaming.nexeditorsdk.nexOverlayKineMasterText.EnableOutline (boolean *enable*)

This method enables the outline effect on the overlay item.

Parameters

enable	`True` if the effect is enabled; otherwise `false`.

Since

version 1.5.23

4.47.3.3 void com.nexstreaming.nexeditorsdk.nexOverlayKineMasterText.EnableShadow (boolean *enable*)

This method enables the shadow effect on the overlay item.

Parameters

enable	`True` if the effect is enabled; otherwise `false`.

Since

version 1.5.23

4.47.3.4 String com.nexstreaming.nexeditorsdk.nexOverlayKineMasterText.getFontId ( )

This method gets the ID of the built-in SDK font set to the current overlay item.

Returns

The ID of the set built-in SDK font nexFont or .ttf file path.

Since

version 1.5.23

4.47.3.5 int com.nexstreaming.nexeditorsdk.nexOverlayKineMasterText.getGlowColor ( )

This method gets the color of the glow set to the current overlay item.

Returns

The color of the glow in the argb form. (e.g. 0xffff0000 ->Red)

Since

version 1.5.23

4.47.3.6 int com.nexstreaming.nexeditorsdk.nexOverlayKineMasterText.getHeight ( )

This method gets the height value when a text is made into a Bitmap.

Returns

The overlay height of the text.

Since

version 1.5.23

4.47.3.7 int com.nexstreaming.nexeditorsdk.nexOverlayKineMasterText.getOutlineColor ( )

This method gets the color of the outline set to the current overlay item.

Returns

The color of the outline in the argb form. (e.g. 0xffff0000 ->Red)

Since

version 1.5.23

4.47.3.8 int com.nexstreaming.nexeditorsdk.nexOverlayKineMasterText.getShadowColor ( )

This method gets the color of the shadow set to the current overlay item.

Returns

The color of the shadow in the argb form. (e.g. 0xffff0000 ->Red)

Since

version 1.5.23

4.47.3.9 String com.nexstreaming.nexeditorsdk.nexOverlayKineMasterText.getText ( )

This method gets the text as a `string` to print on the overlay item.

Returns

The input text.

Since

version 1.5.23

4.47.3.10 int com.nexstreaming.nexeditorsdk.nexOverlayKineMasterText.getTextColor ( )

This method gets the color of the text printed on the current overlay item.

Returns

The color of the text in argb form. (e.g. 0xffff0000 ->Red)

Since

version 1.5.23

4.47.3.11 float com.nexstreaming.nexeditorsdk.nexOverlayKineMasterText.getTextSize ( )

This method gets the size of the text printed on the current overlay item.

Returns

The size of text.

Since

version 1.5.23

4.47.3.12 int com.nexstreaming.nexeditorsdk.nexOverlayKineMasterText.getWidth ( )

This method gets the width value when a text is made into a Bitmap.

Returns

The overlay width of the text.

Since

version 1.5.23

4.47.3.13 boolean com.nexstreaming.nexeditorsdk.nexOverlayKineMasterText.isEnableGlow ( )

This method indicates whether or not the glow effect is enabled on the current overlay item.

Returns

`True` if the effect is on; otherwise `false`.

Since

version 1.5.23

4.47.3.14 boolean com.nexstreaming.nexeditorsdk.nexOverlayKineMasterText.isEnableOutline ( )

This method indicates whether or not the outline effect is enabled on the current overlay item.

Returns

`True` if the effect is on; otherwise `false`.

Since

version 1.5.23

4.47.3.15 boolean com.nexstreaming.nexeditorsdk.nexOverlayKineMasterText.isEnableShadow ( )

This method indicates whether or not the shadow effect is enabled on the current overlay item.

Returns

`True` if the effect is on; otherwise `false`.

Since

version 1.5.23

4.47.3.16 void com.nexstreaming.nexeditorsdk.nexOverlayKineMasterText.setFontId (String *fontId*)

This method sets the built-in SDK font. If this method is not set, the device's Android default font will be used.

Parameters

fontId	The ID of built-in SDK font nexFont or .ttf file path.

Example code : 1

```
File ffile = new File(Environment.getExternalStorageDirectory().getAbsolutePath()+File.separator+"gallery"+
      File.separator+"font.ttf");
 if( ffile.isFile() ) {
     mDefaultFont = ffile.getAbsolutePath();
 }

 mOverlayKMText = new nexOverlayKineMasterText(getApplicationContext(),"", 50);
 mOverlayKMText.setFontId(mDefaultFont);
```

Example code : 2

```
String[] ids = nexFont.getFontIds();
 mOverlayKMText = new nexOverlayKineMasterText(getApplicationContext(),"", 50);
 if(ids.length > 0){
     mOverlayKMText.setFontId(ids[0]);
 }
```

Since

version 1.5.23

4.47.3.17 void com.nexstreaming.nexeditorsdk.nexOverlayKineMasterText.setGlowColor (int *argbColor*)

This method sets the color of the glow set to the current overlay item.

Parameters

argbColor	The color of the glow in the argb form. (e.g. 0xffff0000 ->Red)

Since

version 1.5.23

4.47.3.18 void com.nexstreaming.nexeditorsdk.nexOverlayKineMasterText.setOutlineColor (int *argbColor*)

This method sets the color of the outline of the overlay item.

Parameters

argbColor	The color of the outline in the argb form. (e.g. 0xffff0000 ->Red)

Since

version 1.5.23

4.47.3.19 void com.nexstreaming.nexeditorsdk.nexOverlayKineMasterText.setShadowColor (int *argbColor*)

This method sets the color of the shadow of the overlay item.

Parameters

argbColor	The color of the shadow in the argb form. (e.g. 0xffff0000 ->Red)

Since

version 1.5.23

4.47.3.20 void com.nexstreaming.nexeditorsdk.nexOverlayKineMasterText.setText (String *text*)

This method sets the text to print on the overlay item.

Parameters

text	The text to set.

Since

version 1.5.23

4.47.3.21 void com.nexstreaming.nexeditorsdk.nexOverlayKineMasterText.setTextColor (int *argb*)

This method sets the color of the text to print on the overlay item.

Parameters

argb	The color of the text as in `integer`. (e.g. 0xffff0000 ->Red)

Since

version 1.5.23

4.47.3.22 void com.nexstreaming.nexeditorsdk.nexOverlayKineMasterText.setTextSize (float *size*)

This method sets the size of the text to print on the overlay item.

Parameters

size	The size of text, which follows the size of class `Paint`.

Since

version 1.5.23

4.48 com.nexstreaming.nexeditorsdk.nexOverlayManager Class Reference

Classes

- class nexTitleInfo
- class Overlay

Public Member Functions

- void loadOverlay ()
- List< Overlay > getOverlays ()
- boolean parseOverlay (String overlayId, List< nexTitleInfo > titleInfos)
- boolean applyOverlayToProjectById (nexProject project, String overlayId, List< nexTitleInfo > titleInfos)

Static Public Member Functions

- static nexOverlayManager getOverlayManager (Context appContext, Context resContext)

4.48.1 Detailed Description

This class is used to retrieve and set overlay title templates.

Since

version 1.7.0

4.48.2 Member Function Documentation

4.48.2.1 boolean com.nexstreaming.nexeditorsdk.nexOverlayManager.applyOverlayToProjectById (nexProject *project,* String *overlayId,* List< nexTitleInfo > *titleInfos*)

This method is used to apply an overlay title template to a project with title information.

Parameters

project	Instance of project.
overlayId	ID of overlay title template for applying.
titleInfos	Instance of nexOverlayTitleInfo list for applying title information.

Returns

`TRUE` or `FALSE`.

Since

version 1.7.0

4.48.2.2 static **nexOverlayManager** com.nexstreaming.nexeditorsdk.nexOverlayManager.getOverlayManager (Context *appContext*, Context *resContext*) [static]

This method is used to get a nexOverlayManager instance.

Parameters

appContext	The application context instance.
resContext	The context instance that can access the asset.

Returns

The nexOverlayManager instance.

Since

version 1.7.0

4.48.2.3 List< **Overlay** > com.nexstreaming.nexeditorsdk.nexOverlayManager.getOverlays ( )

This method is used to get all installed overlay title templates.

Returns

The list of installed overlay title templates.

Since

version 1.7.0

4.48.2.4 void com.nexstreaming.nexeditorsdk.nexOverlayManager.loadOverlay ( )

This method is used to load all installed overlay title templates.

Since

version 1.7.0

4.48.2.5 boolean com.nexstreaming.nexeditorsdk.nexOverlayManager.parseOverlay (String *overlayId*, List< **nexTitleInfo** > *titleInfos*)

This method is used to parse the title information list from an overlay title template.

Parameters

overlayId	ID of an overlay title template for parsing.
titleInfos	Instance of the nexOverlayTitleInfo list for receiving title information.

Returns

TRUE or FALSE.

Since

version 1.7.0

4.49 com.nexstreaming.nexeditorsdk.nexOverlayPreset Class Reference

Public Member Functions

- nexOverlayImage getOverlayImage (String overlayImageId)
- String[] getIDs ()
- Bitmap getIcon (String overlayImageId, int width, int height)

  This method gets the main thumbnail image of an Asset Sticker. If the sticker format is SVG, this gets the thumbnail of a desired size.

Static Public Member Functions

- static nexOverlayPreset getOverlayPreset (Context context)
- static nexOverlayPreset getOverlayPreset ()

4.49.1 Detailed Description

This class allows the application to access image resources in the `assets/overlay` folder.

Returns

A bitmap image.

Since

version 1.5.19

4.49.2 Member Function Documentation

4.49.2.1 Bitmap com.nexstreaming.nexeditorsdk.nexOverlayPreset.getIcon (String *overlayImageId,* int *width,* int *height*)

This method gets the main thumbnail image of an Asset Sticker. If the sticker format is SVG, this gets the thumbnail of a desired size.

Parameters

overlayImageId	Item ID
width	The desired width of a thumbnail.
height	The desired height of a thumbnail.

Returns

The bitmap of a thumbnail.

Since

1.7.0

4.49.2.2 String [ ] com.nexstreaming.nexeditorsdk.nexOverlayPreset.getIDs ( )

This method gets the resource IDs in the `assets/overlay` folder.

Example code :

```
String[] overlayIds = overlayPreset.getIDs();
```

Returns

The resource IDs, as a `String`.

Since

version 1.5.15

4.49.2.3 nexOverlayImage com.nexstreaming.nexeditorsdk.nexOverlayPreset.getOverlayImage (String *overlayImageId*)

This method returns a `nexOverlayImage` instance, which corresponds to the value set by the parameter.

Example code :

```
String[] overlayIds = overlayPreset.getIDs();
for(String overlayId: overlayIds) {
    if(overlayId.equals("search_id"))
        nexOverayImage overlayImage = ovelayPreset.getOverlayImage(overlayId);
    ...
}
```

Parameters

overlayImageId	An ID of `nexOverlayImage`.

Returns

An instance of `nexOverlayImage`.

Since

version 1.5.15

4.49.2.4 static nexOverlayPreset com.nexstreaming.nexeditorsdk.nexOverlayPreset.getOverlayPreset (Context *context*) [static]

This method returns a nexOverlayPreset instance.

Parameters

context	The context to get asset.

Returns

An instance of nexOverlayPreset.

Since

version 1.5.15

4.49.2.5 static nexOverlayPreset com.nexstreaming.nexeditorsdk.nexOverlayPreset.getOverlayPreset ( ) [static]

This method returns a nexOverlayPreset instance.

Returns

An instance of nexOverlayPreset.

Since

version 1.5.15

4.50 com.nexstreaming.nexeditorsdk.nexEngine.nexPlayState Enum Reference

Public Member Functions

- **nexPlayState** (int value)
- int **getValue** ()

Static Public Member Functions

- static nexPlayState **fromValue** (int value)

Public Attributes

- **NONE** =(0)
- **IDLE** =(1)
- **RUN** =(2)
- **RECORD** =(3)

4.50.1 Detailed Description

This enumeration defines the possible state values of the `nexEngine`.

Since

version 1.3.43

4.51 com.nexstreaming.nexeditorsdk.nexProject Class Reference

Inherits com.nexstreaming.nexeditorsdk.nexObserver.

Public Member Functions

- nexProject ()
- nexProject (nexTheme Theme)
- nexProject (String ThemeId)
- nexProject (String ThemeId, String open, String end)
- void setTitle (String open, String end)
- String getOpeningTitle ()
- String getEndingTitle ()
- List< nexClip > getPrimaryItems ()
- List< nexClip > getSecondaryItems ()
- List< nexAudioItem > getAudioItems ()
- int getTotalClipCount (boolean bPrimary)
- int getTotalVisualClipCount ()
- int add (nexClip item)
- int add (int index, boolean bPrimary, nexClip item)
- int addAudio (nexClip clip, int startTime, int endTime)
- int changeAudio (nexAudioItem audioItem, int startTime, int endTime)
- nexAudioItem getAudioItem (int index)
- nexAudioItem findAudioItem (int id)
- void setThemeId (String ThemeId)
- boolean setBackgroundMusicPath (String strMusicPath)
- nexClip getBackgroundMusic ()
- String getBackgroundMusicPath ()
- void setBackgroundConfig (int startTime, boolean useThemeMusic, boolean loop)
- void setBackgroundTrim (int startTime, int endTime)
- String getThemeId ()
- nexClip getClip (int location, boolean bPrimary)
- nexClip getLastPrimaryClip ()

  This method returns the last clip of a video clip track.
- nexClip getNextClip (nexClip clip)
- void move (int index, nexClip item)

- int remove (nexClip item)
- void allClear (boolean primary)
- void setBGMMasterVolumeScale (float scale)
- float getBGMMasterVolumeScale ()
- int getTotalTime ()
- void setTemplateApplyMode (boolean flag)

    This method is called to change the template apply mode flag. This method is used internally in nexTemplate.
- boolean updateProject ()
- void setProjectAudioFadeInTime (int fadeIn)
- void **setProjectVolume** (int volume)
- int **getProjectVolume** ()
- void **setManualVolumeControl** (int manual_vol_ctl)
- int **getManualVolumeControl** ()
- void setProjectAudioFadeOutTime (int fadeOut)
- int getProjectAudioFadeInTime ()
- int getProjectAudioFadeOutTime ()
- boolean addOverlay (nexOverlayItem overlay)
- nexOverlayItem getOverlay (int overlayId)
- boolean removeOverlay (int overlayId)
- void clearOverlay ()
- List< nexOverlayItem > getOverlayItems ()
- int[] getClipDurationTimeGuideLine (int clipIndex)
- int[] getTransitionDurationTimeGuideLine (int clipIndex, int showClipDurationTime)
- int[] getClipPositionTime (int clipIndex)
- int getClipPosition (int projectCTS)
- int getProjectTime2ClipTimePosition (int position, int projectCTS)
- int getClipMaxSpeedControlList (int position)
- Intent makeKineMasterIntent ()

Static Public Member Functions

- static nexProject clone (nexProject src)

Static Public Attributes

- final static int kAutoThemeClipDuration = 6000
- final static int kAutoThemeTransitionDuration = 2000

4.51.1 Detailed Description

This class manages clips to be edited with NexEditor™ SDK as projects.

To edit with the NexEditor™ SDK, a project is created, and then clips (video, image, or audio) are added to the project and further edited with the options available.

A finished project is then exported to become the final version of a NexEditor™ video.

The following sample code demonstrates how to create a project and add clips to the project: **class sample code**

```
nexProject project = new nexProject(mThemeID,"Open","Close");
  for(int i=m_listfilepath.size()-1; i>=0; i--) {
      nexClip clip = nexClip.getSupportedClip(m_listfilepath.get(i));
      if( clip != null )
          project.add(clip);
  }
  String transitionID = "nexstring..XXXXXX";
  project.getClip(0,true).getTransitionEffect().setTransitionEffect(transitionID);
```

Since

version 1.0.0

4.51.2 Constructor & Destructor Documentation

4.51.2.1 com.nexstreaming.nexeditorsdk.nexProject.nexProject ( )

This method creates a project using the basic theme.

Example code :

```
private nexProject project = new nexProject();
```

Since

version 1.0.0

4.51.2.2 com.nexstreaming.nexeditorsdk.nexProject.nexProject (nexTheme *Theme*)

This method creates an instance of a project using a chosen `Theme`.

In NexEditor™ SDK, a theme includes background music as well as opening and closing title and related transition effects to create a visually cohesive video experience for a user who doesn't want to set each individual option when editing.

Example code :

```
mProject = new nexProject(EffectLibrary.getEffectLibrary(getApplicationContext()).findThemeById(
      com.nexstreaming.kinemaster.travel));
```

Parameters

Theme	The theme to set for the project being created.

Since

version 1.0.0

4.51.2.3 com.nexstreaming.nexeditorsdk.nexProject.nexProject (String *ThemeId*)

This method creates an instance of a project using the received theme ID from the `ThemeId` parameter.

Example code :

```
private nexProject project = new nexProject("com.nexstreaming.kinemaster.travel");
```

Parameters

ThemeId	The ID of the theme to set for the project being created, as a `String`.

Since

version 1.0.0

4.51.2.4 com.nexstreaming.nexeditorsdk.nexProject.nexProject (String *ThemeId,* String *open,* String *end*)

The generator of NexEditor™ projects.

Example code :

```
nexProject project = new nexProject(mThemeID,"Open my Create","End my Close");
```

Parameters

ThemeId	The theme ID of the project as a `String`.
open	The text title that appears at the opening of the video project.
end	The text title that appears at the end of the video project.

Since

version 1.0.0

4.51.3 Member Function Documentation

4.51.3.1 int com.nexstreaming.nexeditorsdk.nexProject.add (nexClip *item*)

This method adds a clip to a track.

Example code :

```
nexClip clip = nexClip.getSolidClip(Color.RED);
if( clip != null ) {
    project.add(clip);
}
```

Parameters

item	The nexClip instance to be added.

Returns

0 if successful; otherwise a RuntimeException.

Since

version 1.0.0

4.51.3.2 int com.nexstreaming.nexeditorsdk.nexProject.add (int *index,* boolean *bPrimary,* nexClip *item*)

This method adds a clip to a track.

Example code :

```
nexClip clip = nexClip.getSolidClip(Color.RED);
if( clip != null ) {
    project.add(index, true, clip);
}
```

Parameters

index	The index of the clip to be added.
bPrimary	Set to TRUE if the track is composed of video clips and image clips only; set to FALSE if the track is composed of audio clips only.
item	The nexClip instance to be added.

Returns

0 if successful; otherwise a RuntimeException.

Since

version 1.0.0

4.51.3.3 int com.nexstreaming.nexeditorsdk.nexProject.addAudio (nexClip *clip,* int *startTime,* int *endTime*)

This method adds an audio clip to a project.

Example code :

```
project.addAudio(clip2, 0, clip2.getTotalTime());
```

Parameters

clip	The type of clip to add: in this case `audio`.
startTime	The start time of the audio clip to be added (relative to the timeline of the project).
endTime	The end time of the audio clip to be added.

Returns

Always 0.

Since

version 1.0.0

4.51.3.4 boolean com.nexstreaming.nexeditorsdk.nexProject.addOverlay (nexOverlayItem *overlay*)

This method adds an overlay item to a project.

Example code :

```
nexAnimate ani = nexAnimate.getAnimateImages(
    0, 5000, R.drawable.ilove0001
    , R.drawable.ilove0002
    , R.drawable.ilove0003
    , R.drawable.ilove0004
    , R.drawable.ilove0005
    , R.drawable.ilove0006
    , R.drawable.ilove0007
    , R.drawable.ilove0008
    , R.drawable.ilove0009
    , R.drawable.ilove0010
    );
overlayItem.addAnimate(ani);
mProject.addOverlay(overlayItem);
```

Parameters

overlay	The overlay item to add to a project.

Returns

`TRUE` if the overlay ID exists and the overlay item is added; otherwise `FALSE`.

See Also

removeOverlay(int)
getOverlay(int)

Since

version 1.1.0

4.51.3.5 void com.nexstreaming.nexeditorsdk.nexProject.allClear (boolean *primary*)

This method clears all of the clips in a track in a project.

Example code :

```
mProject.allClear(true);
```

Parameters

primary	Set to `TRUE` if the track is composed of video clips or image clips only; Otherwise set to `FALSE` if the track is composed of audio clips only.

Since

version 1.0.0

4.51.3.6 int com.nexstreaming.nexeditorsdk.nexProject.changeAudio (nexAudioItem *audioItem,* int *startTime,* int *endTime*)

This method allows to change the start time and the end time of an audio item.

Example code :

```
project.changeAudio(project.getAudioItem(0), mStartTime, project.getTotalTime());
```

Parameters

audioItem	An audio item to change the start time and end time.
startTime	The new start time of the audio item.
endTime	The new end time of the audio item.

Returns

-1 after the sorting if the audio item needs to be sorted, otherwise 0.

See Also

addAudio(nexClip, int, int)

Since

version 1.3.52

4.51.3.7 void com.nexstreaming.nexeditorsdk.nexProject.clearOverlay ( )

This method removes all of the overlay items from a project.

To remove a specific overlay item from a project, please call the `removeOverlay()` method with the overlay ID of the item instead.

Example code :

```
mProject.clearOverlay();
```

See Also

removeOverlay(int)

Since

version 1.1.0

4.51.3.8 static nexProject com.nexstreaming.nexeditorsdk.nexProject.clone (nexProject *src*) [static]

This method duplicates a project. The duplicated project is an independent project from the original project it was copied from.

Example code :

```
mExportProject = nexProject.clone(mDisplayProject);
```

Parameters

src	A project to duplicate.

Returns

A duplicated new project.

Since

version 1.3.43

4.51.3.9 nexAudioItem com.nexstreaming.nexeditorsdk.nexProject.findAudioItem (int *id*)

This method gets an audio item using the ID set with the parameter `id`.

Example code :

```
project.findAudioItem(id).setTrim(mStartTrim, mEndTrim);
```

Parameters

id	ID of an audio item.

Returns

nexAudioItem

Since

version 1.3.52

4.51.3.10 nexAudioItem com.nexstreaming.nexeditorsdk.nexProject.getAudioItem (int *index*)

This method gets an audio item, from the audio item list, by the parameter `index`.

Example code :

```
project.changeAudio(project.getAudioItem(0), mStartTime, project.getTotalTime());
```

Parameters

index	index of an audio item.

Returns

nexAudioItem

Since

version 1.3.52

4.51.3.11 List<nexAudioItem> com.nexstreaming.nexeditorsdk.nexProject.getAudioItems ( )

This method gets a list of the audio items that can be trimmed and have their volume adjusted.

Example code :

```
List<nexAudioItem> alist =  mProject.getAudioItems();
```

Returns

A list of the audio items, `nexAudioItem`, that can be trimmed and volume-adjusted.

Since

version 1.1.0

4.51.3.12 nexClip com.nexstreaming.nexeditorsdk.nexProject.getBackgroundMusic ( )

This method gets the background music for a project.

Example code :

```
nexClip bgm = mProject.getBackgroundMusic();
```

Returns

The nexClip instance of the background music.

Since

version 1.0.0

4.51.3.13 String com.nexstreaming.nexeditorsdk.nexProject.getBackgroundMusicPath ( )

This method gets the path of the background music for a project.

Example code :

```
path = project.getBackgroundMusicPath();
```

Returns

The location where the background music audio file is saved.

Since

version 1.0.0

4.51.3.14 float com.nexstreaming.nexeditorsdk.nexProject.getBGMMasterVolumeScale ( )

This method gets the master volume of the background music set to a project.

Example code :

```
item.mClipVolume = (int) (mProject.getBGMMasterVolumeScale() * 200);
```

Returns

The value of the background music master volume.

Since

version 1.3.4

4.51.3.15 nexClip com.nexstreaming.nexeditorsdk.nexProject.getClip (int *location,* boolean *bPrimary*)

This method gets a clip in a project.

This method can be used to get any kind of clip included in a project, including a video, an image, or an audio clip.

Example code :

```
project.getClip(0, false).setClipVolume(mBGMVol);
```

Parameters

location	The location of the clip.
bPrimary	Set to `TRUE` if the track is composed of video and image clips; otherwise set to `FALSE`.

Returns

The `nexClip` instance from the project.

Since

version 1.0.0

4.51.3.16 int [] com.nexstreaming.nexeditorsdk.nexProject.getClipDurationTimeGuideLine (int *clipIndex*)

This method guides for a clip's best time range to be added in the primary track of a project.

For example, if a transition effect takes 5 seconds whilst a clip(video and image) takes 3 seconds, the clip can't be viewed properly because the transition effect is longer. To avoid this, this method guides for the best duration of a clip to be played properly in the project.

Example code :

```
int[] timeline = mProject.getClipDurationTimeGuideLine(index);
```

Parameters

clipIndex	The index of a clip from the primary track of the project.

Returns

The minimum clip duration on index 0 and maximum clip duration on index 1 as `msec` (milliseconds), in `array`.

Since

version 1.3.0

4.51.3.17 int com.nexstreaming.nexeditorsdk.nexProject.getClipMaxSpeedControlList (int *position*)

Gets the maximum speed control value applicable to the video clip in the project.

Parameters

position	The clip position in the project.

Returns

The maximum speed control value that can be applied(13 ∼ 400).

Since

1.5.42

4.51.3.18 int com.nexstreaming.nexeditorsdk.nexProject.getClipPosition (int *projectCTS*)

This method gets the index of a clip on the specific location of project's timeline.

Example code :

```
int position = project.getClipPosition(2000);
```

Parameters

projectCTS	The specific location of the project's timeline in `msec` (milliseconds) as `integer`.

Returns

The index of a clip located on the project's timeline specified by parameter `projectCTS`. -1 if the index can not be found.

See Also

getProjectTime2ClipTimePosition(int, int)

Since

version 1.3.43

4.51.3.19 int [] com.nexstreaming.nexeditorsdk.nexProject.getClipPositionTime (int *clipIndex*)

This method gets the location, on the timeline, of a clip(video and image) on the primary track of a project.

Example code :

```
int[] clipPositionTime = mProject.getClipPositionTime(clipindex);
```

com.nexstreaming.nexeditorsdk.nexClip#getProjectStartTime() of the class `nexClip` can be used instead of this method.

Parameters

clipIndex	The index of a clip on the primary track.

Returns

The start time on index 0 and end time on index 1 as `msec` (milliseconds), in `array`.

Since

version 1.3.0

4.51.3.20 String com.nexstreaming.nexeditorsdk.nexProject.getEndingTitle ( )

This method get the ending title of a project.

Example code :

```
strTitle = mProject.getEndingTitle();
```

Returns

The text set for the theme ending of a project.

Since

version 1.0.0

4.51.3.21 nexClip com.nexstreaming.nexeditorsdk.nexProject.getLastPrimaryClip ( )

This method returns the last clip of a video clip track.

Returns

The last clip

Since

1.7.0

4.51.3.22 nexClip com.nexstreaming.nexeditorsdk.nexProject.getNextClip (nexClip *clip*)

Given a clip already in a project, this method returns the clip that immediately follows that reference clip.

Example code :

```
nextClip = project.getNextClip(project.getClip(0, false));
```

Parameters

clip	The reference clip from the project.

Returns

The clip following the reference clip passed with the parameter `clip`.

Since

version 1.0.0

4.51.3.23 String com.nexstreaming.nexeditorsdk.nexProject.getOpeningTitle ( )

This method gets the opening title of a project.

Example code :

```
strTitle = mProject.getOpeningTitle();
```

Returns

The title set for the theme opening of a project.

Since

version 1.0.0

4.51.3.24 nexOverlayItem com.nexstreaming.nexeditorsdk.nexProject.getOverlay (int *overlayId*)

This method gets an overlay item from its corresponding overlay ID.

Example code :

```
mEngine.getProject().getOverlay(mOverlayId).setPosition(lastProgress, mEngine.getProject().
      getOverlay(mOverlayId).getPositionY());
```

Parameters

overlayId	The ID of the overlay item to retrieve.

Returns

The corresponding overlay item if the overlay ID passed exists; otherwise `null`.

See Also

addOverlay(nexOverlayItem)

Since

version 1.1.0

4.51.3.25 List<**nexOverlayItem**> com.nexstreaming.nexeditorsdk.nexProject.getOverlayItems ( )

This method returns a list of all of the overlay items in a project.

Example code :

```
mSelectedPIP2.addAll(mExportProject.getOverlayItems());
```

Returns

The overlay items in a project, as a list of `nexOverlayItem` instances.

Since

version 1.1.0

4.51.3.26 List<**nexClip**> com.nexstreaming.nexeditorsdk.nexProject.getPrimaryItems ( )

This method gets a read-only list of clips, from the track with video clips and image clips only. A project contains two different kinds of tracks: one track for video and image clips and other tracks for audio clips.

Example code :

```
List<nexClip> list = mProject.getPrimaryItems();
```

Returns

The list of `nexClip` clips, that are video or image clips.

Since

version 1.0.0

4.51.3.27 int com.nexstreaming.nexeditorsdk.nexProject.getProjectAudioFadeInTime ( )

This method gets the time to take for project audio to fade-in at the beginning of a project.

Example code :

```
int fadeIn = mProject.getProjectAudioFadeInTime();
```

Returns

The time it takes for the audio to fade-in, in `msec` (milliseconds).

See Also

setProjectAudioFadeInTime(int)

Since

version 1.0.0

4.51.3.28 int com.nexstreaming.nexeditorsdk.nexProject.getProjectAudioFadeOutTime ( )

This method gets the time over which the volume fades out at the end of a playing project.

Example code :

```
int fadeOut = mProject.getProjectAudioFadeOutTime();
```

Returns

The time it takes for the audio to fade out at the end of a project, in `msec` (milliseconds).

See Also

setProjectAudioFadeOutTime(int)

Since

version 1.0.0

4.51.3.29 int com.nexstreaming.nexeditorsdk.nexProject.getProjectTime2ClipTimePosition (int *position,* int *projectCTS*)

This method gets time of a clip when the clip is on the specific location on the timeline specified by the parameter `position`. For example, when there are three(3) videos with duration of 4 seconds each in a project, and the input value for `position` is 5, this method will return 1 because 5 second on the timeline will be on the 1 second on the second video.

Example code :

```
int time = project.getProjectTime2ClipTimePosition(position, 2000);
```

Parameters

position	The index of a clip. Returned value from the method `getClipPosition()`.
projectCTS	The specific location of the project's timeline in `msec` (milliseconds) as `integer`.

Returns

position The time of a clip corresponding the specified time by the parameter `projectCTS` on the timeline.

See Also

getClipPosition(int)

Since

version 1.3.43

4.51.3.30 List<**nexClip**> com.nexstreaming.nexeditorsdk.nexProject.getSecondaryItems ( )

This method gets a read-only list from the track with audio clips only. A project contains two different kinds of tracks: one track for video and image clips and other tracks for audio clips.

Returns

The list of `nexClip` audio clips.

Since

version 1.0.0

Deprecated For internal use only. Please do not use.

4.51.3.31 String com.nexstreaming.nexeditorsdk.nexProject.getThemeId ( )

This method gets the ID of the theme that is set for a project.

Example code :

```
bgm = nexClip.getSupportedClip(mEffectLibrary.getMusicForTheme(mProject.getThemeId()).getAbsolutePath());
```

Returns

The theme ID currently set for the project.

Since

version 1.0.0

4.51.3.32 int com.nexstreaming.nexeditorsdk.nexProject.getTotalClipCount (boolean *bPrimary*)

This method gets the total number of clips saved in a specific track.

Example code :

```
for(int i = 0; i<mProject.getTotalClipCount(false); i++){
    if(duration < mProject.getAudioItem(i).getEndTime()){
        duration += mProject.getAudioItem(i).getEndTime();
    }
}
```

Parameters

bPrimary	Set to TRUE if the track is composed of video clips and image clips only; set to FALSE if the track is composed of audio clips only.

Returns

The number of clips in the selected track, as an integer.

Since

version 1.0.0

4.51.3.33 int com.nexstreaming.nexeditorsdk.nexProject.getTotalTime ( )

This method gets the total playtime of a project. This method can be used instead of nexEngine#getDuration() of nexEngine.

Example code :

```
progressBar_playtime.setMax(project.getTotalTime() / 1000);
```

Returns

The total playtime of the project in msec (milliseconds).

See Also

nexEngine::getDuration()

Since

version 1.0.0

4.51.3.34 int com.nexstreaming.nexeditorsdk.nexProject.getTotalVisualClipCount ( )

This method gets the total number of image clips and video clips in a project.

This includes the number of trimmed video clips. For example if there is one image clip and video clips that have been divided and trimmed into three parts, then 1 + 3 = 4 will be returned.

Returns

The total number of visual clips in the project, as an `integer`.

Since

version 1.0.0

Deprecated For internal use only. Please do not use.

4.51.3.35 int [ ] com.nexstreaming.nexeditorsdk.nexProject.getTransitionDurationTimeGuideLine (int *clipIndex,* int *showClipDurationTime*)

This method guides for the best duration of a transition effect, when applied to the clips in a project.

Example code :

```
int[] TrTimeLine = mProject.getTransitionDurationTimeGuideLine(clipindex, 2000);
```

Parameters

clipIndex	The index of the clips on the primary track of the project.
showClipDurationTime	The minimum duration of a transition effect in `msec` (milliseconds).

Returns

The minimum effect duration on index 0 and maximum effect duration on index 1 as `msec` (milliseconds), in `array`.

Since

version 1.3.0

4.51.3.36 Intent com.nexstreaming.nexeditorsdk.nexProject.makeKineMasterIntent ( )

This method shares the project that had been edited in NexEditor™ .

Deprecated For internal use only. Please do not use.

Warning

Use this method only when KineMaster™ has been installed, which is compatible with NexEditor™ .

Example code :

```
startActivity(mEngin.getProject().makeKineMasterIntent());
```

Returns

The intent that can call KineMaster™ .

Since

1.5.24

4.51.3.37 void com.nexstreaming.nexeditorsdk.nexProject.move (int *index*, nexClip *item*)

This method moves a clip in a project.

Example code :

```
mProject.move(pos - 1, mProject.getClip(pos, true));
```

Parameters

index	The location where the clip will be moved.
item	The nexClip to be moved.

Since

version 1.0.0

4.51.3.38 int com.nexstreaming.nexeditorsdk.nexProject.remove (nexClip *item*)

This method removes a clip from a project.

Example code :

```
project1.remove(clip);
```

Parameters

item	The nexClip to be removed.

Since

version 1.0.0

4.51.3.39 boolean com.nexstreaming.nexeditorsdk.nexProject.removeOverlay (int *overlayId*)

This method removes an overlay item from a project.

Example code :

```
mExportProject.removeOverlay(mSelectedPIP2.get(2).getId());
```

Note

To remove all overlay items in a project, please call the clearOverlay() method instead.

Parameters

overlayId	The overlay ID of the overlay item to remove, as an integer.

Returns

TRUE if the overlay item is successfully removed; otherwise FALSE.

See Also

clearOverlay()

Since

version 1.1.0

4.51.3.40 void com.nexstreaming.nexeditorsdk.nexProject.setBackgroundConfig (int *startTime,* boolean *useThemeMusic,* boolean *loop*)

This method sets additional settings on the background music. This method should be called first before calling nexEngine#updateProject() to successfully apply the settings value.

Example code :

```
project.setBackgroundConfig(mStartTime, false, loop.isChecked());
```

Parameters

startTime	The start time of the background music within the project as an `integer`.
useThemeMusic	Set to `TRUE` to enable theme music (if a theme was set to the project) when the background music (set from the user's own music file) is not set or unavailable; otherwise `FALSE`.
loop	Set to `TRUE` to loop the background music; otherwise `FALSE`.

Since

version 1.3.4

4.51.3.41 boolean com.nexstreaming.nexeditorsdk.nexProject.setBackgroundMusicPath (String *strMusicPath*)

This method sets the background music of a project.

Background music for a project plays throughout the entire project.

Example code :

```
project.setBackgroundMusicPath(m_soundfilepath);
```

Parameters

strMusicPath	The location where the background music file is saved.

Returns

`TRUE` if the audio file is supported by NexEditor™ , otherwise `FALSE`.

Since

version 1.0.0

4.51.3.42 void com.nexstreaming.nexeditorsdk.nexProject.setBackgroundTrim (int *startTime,* int *endTime*)

This method set the BGM trim

Parameters

startTime	BGM trim start time
endTime	BGM trim end time

Since

version 1.5.15

4.51.3.43 void com.nexstreaming.nexeditorsdk.nexProject.setBGMMasterVolumeScale (float *scale*)

This method sets the master volume of the background music set to a project.

Example code :

```
project.setBGMMasterVolumeScale(mBGMVol / 100f);
```

Parameters

scale	The master volume in scale from 0 to 1 where 1 is the original volume.

Since

version 1.3.4

4.51.3.44 void com.nexstreaming.nexeditorsdk.nexProject.setProjectAudioFadeInTime (int *fadeIn*)

This method sets the fade-in time for the audio at the beginning of a project.

This method allows the volume of audio to gradually increase at the beginning of the project, making the addition of any types of audio less jarring and smoother.

The default fade-in time for project audio is 200 milliseconds.

Example code :

```
project.setProjectAudioFadeInTime(Integer.parseInt(fadeIn.getText().toString()));
project.setProjectAudioFadeOutTime(Integer.parseInt(fadeOut.getText().toString()));
```

Parameters

fadeIn	The time it takes for the audio to fade-in, in `msec` (milliseconds).

See Also

setProjectAudioFadeOutTime

Since

version 1.0.0

4.51.3.45 void com.nexstreaming.nexeditorsdk.nexProject.setProjectAudioFadeOutTime (int *fadeOut*)

This method gradually decreases the volume of audio at the end of a playing project.

The default fade-out time is 5000 milliseconds.

Example code :

```
project.setProjectAudioFadeInTime(Integer.parseInt(fadeIn.getText().toString()));
project.setProjectAudioFadeOutTime(Integer.parseInt(fadeOut.getText().toString()));
```

Parameters

fadeOut	The time it takes for the audio to fade-out, in `msec` (milliseconds).

See Also

setProjectAudioFadeInTime

Since

version 1.0.0

4.51.3.46 void com.nexstreaming.nexeditorsdk.nexProject.setTemplateApplyMode (boolean *flag*)

This method is called to change the template apply mode flag. This method is used internally in nexTemplate.

Example code :

```
project.setTemplateApplyMode();
```

Parameters

flag	template apply mode flag.

Since

version 1.5.50

4.51.3.47 void com.nexstreaming.nexeditorsdk.nexProject.setThemeId (String *ThemeId*)

This method changes the theme of a project.

Example code :

```
project.setThemeId("com.nexstreaming.kinemaster.travel");
```

Parameters

ThemeId	The theme ID of the new theme to be changed, as a `String`.

Since

version 1.0.0

4.51.3.48 void com.nexstreaming.nexeditorsdk.nexProject.setTitle (String *open,* String *end*)

This method sets the opening and ending titles of a project.

Example code :

```
project.setTitle("Open my Create","End my Close");
```

Parameters

open	The text title that appears at the opening of the video.
end	The text title that appears at the end of the video.

Since

version 1.0.0

4.51.3.49 boolean com.nexstreaming.nexeditorsdk.nexProject.updateProject ( )

This method is called to get an update of the total playtime of a project when changes are made on a track. This method is used internally by getTotalTime() , getClipPositionTime(int), and nexEngine#updateProject().

Example code :

```
project.updateProject();
```

See Also

getTotalTime()
getClipPositionTime(int)
nexEngine::updateProject()

Since

version 1.0.0

4.51.4 Member Data Documentation

4.51.4.1 final static int com.nexstreaming.nexeditorsdk.nexProject.kAutoThemeClipDuration = 6000 [static]

The time to set as the automatic duration for image clips in a particular theme.

Since

version 1.0.0

4.51.4.2 final static int com.nexstreaming.nexeditorsdk.nexProject.kAutoThemeTransitionDuration = 2000 [static]

The time to set as the automatic duration of a transition effect applied between project clips.

Since

version 1.0.0

4.52 com.nexstreaming.nexeditorsdk.nexTemplate Class Reference

Public Member Functions

- nexTemplate ()
- void release ()
- ArrayList< nexTemplateTitle > **getTitleInfo** ()
- void **setTitleInfo** (ArrayList< nexTemplateTitle > titleInfo)

4.52.1 Constructor & Destructor Documentation

4.52.1.1 com.nexstreaming.nexeditorsdk.nexTemplate.nexTemplate ( )

This method creates an instance of `nexTemplateComposer`.

Since

version 1.0.0

4.52.2 Member Function Documentation

4.52.2.1 void com.nexstreaming.nexeditorsdk.nexTemplate.release ( )

This method clears the resources of `nexTemplateComposer`.

This method must be called when the activity is destroyed.

Since

version 1.0.0

4.53 com.nexstreaming.nexeditorsdk.nexTemplateComposer Class Reference

Public Member Functions

- nexTemplateComposer ()

- nexProject createProject ()
- void release ()
- String setTemplateEffects2Project (nexProject project, Context appContext, Context resContext, String templateFilePath, boolean updateState, ArrayList< nexTemplateTitle > titleInfo) throws nexSDKException
- ArrayList< nexTemplateTitle > parseTitleOfTemplate (Context appContext, Context resContext, String templateFilePath) throws nexSDKException
- boolean updateTitleInfoOnProject (nexProject project, ArrayList< nexTemplateTitle > titleInfo) throws nexSDKException

4.53.1 Detailed Description

This class newly composes clips of a project that has been created from a pre-registered template file.

Since

version 1.0.0

4.53.2 Constructor & Destructor Documentation

4.53.2.1 com.nexstreaming.nexeditorsdk.nexTemplateComposer.nexTemplateComposer ( )

This method creates an instance of `nexTemplateComposer`.

Since

version 1.0.0

4.53.3 Member Function Documentation

4.53.3.1 nexProject com.nexstreaming.nexeditorsdk.nexTemplateComposer.createProject ( )

This method returns an instance after creating an empty project.

Since

version 1.0.0

4.53.3.2 ArrayList<nexTemplateTitle> com.nexstreaming.nexeditorsdk.nexTemplateComposer.parseTitleOfTemplate (Context *appContext,* Context *resContext,* String *templateFilePath*) throws nexSDKException

This method generates the title information of a template `templateFilePath`.

Example code :

```
nexTemplateComposer composer = new nexTemplateComposer();
        ...
        ArrayList<nexTemplateTitle> titleInfo = composer.parseTitleOfTemplate(getApplicationContext(), null
      , tempalte_file_path);
```

Parameters

appContext	The application context instance.
resContext	The context instance that can access the asset.
templateFilePath	The directory where the template file is saved.

Returns

Template title information array list.

Since

version 1.5.48

4.53.3.3 void com.nexstreaming.nexeditorsdk.nexTemplateComposer.release ( )

This method clears the resources of `nexTemplateComposer`. This method must be called when the activity is destroyed.

Since

version 1.0.0

4.53.3.4 String com.nexstreaming.nexeditorsdk.nexTemplateComposer.setTemplateEffects2Project (nexProject *project*, Context *appContext*, Context *resContext*, String *templateFilePath*, boolean *updateState*, ArrayList< nexTemplateTitle > *titleInfo*) throws nexSDKException

This method creates a new project with template files stated by the parameter `templateFilePath`.

Example code :

```
 nexTemplateComposer composer = new nexTemplateComposer();
nexProject project = composer.createProject();
// add clips in the project
...
composer.setTemplateEffects2Project(project, getApplicationContext(), null, tempalte_file_path, false,
      titleInfo);
```

Parameters

project	The nexProject instance.
appContext	The application context instance.
resContext	The context instance that can access the asset.
templateFilePath	The directory where the template file is saved.
updateState	Set to `true` and a project will be created with previous template list. Set to `false` and a project will be created with new template list. Default value is `false`.
titleInfo	The template title information array list from user input Added parameter from version 1.5.48

Since

version 1.0.0

4.53.3.5 boolean com.nexstreaming.nexeditorsdk.nexTemplateComposer.updateTitleInfoOnProject (**nexProject** *project,* ArrayList< **nexTemplateTitle** > *titleInfo*) throws nexSDKException

This method updates the title information of a project.

Example code :

```
nexTemplateComposer composer = new nexTemplateComposer();
              ...
              ArrayList<nexTemplateTitle> titleInfo = composer.parseTitleOfTemplate(getApplicationContext()
      , null, tempalte_file_path);
              composer.setTemplateEffects2Project(project, getApplicationContext(), null,
      tempalte_file_path, false, null);

              // update title with user input.

              boolean err = composer.updateTitleInfoOnProject(project, titleInfo);
              if( err == false )
              {
                  // error case code
              }
```

Parameters

project	The nexProject instance.
titleInfo	The template title information array list from user input Added parameter from version 1.5.48

Returns

Update status.

Since

version 1.5.48

4.54 com.nexstreaming.nexeditorsdk.nexTemplateManager Class Reference

This class provides an easy way to use items of the template Category in nexAssetPackageManager in nexTemplateManager. If the developer uses only templates of nexEditorSDK or separates templates, they can choose not to use nexAssetPackageManager and compose an application with just nexTemplateManager.

Classes

- class Template

Public Member Functions

- List< Template > getTemplates ()
- int[] getTemplateAssetIdxs ()
- String[] getTemplateAssetIds ()
- Template getTemplateById (String id)
- void loadTemplate ()
- boolean applyTemplateToProjectById (nexProject project, String templateId)
- String getLastErrorMessage ()

- String[] getTemplateIds ()
- void uninstallPackageById (String templateId)
- int findNewPackages ()
- boolean isInstallingPackages ()
- void installPackagesAsync (final nexAssetPackageManager.OnInstallPackageListener listener)

Static Public Member Functions

- static nexTemplateManager getTemplateManager (Context appContext, Context resContext)
- static nexTemplateManager getTemplateManager ()

4.54.1 Detailed Description

This class provides an easy way to use items of the template Category in nexAssetPackageManager in nexTemplateManager. If the developer uses only templates of nexEditorSDK or separates templates, they can choose not to use nexAssetPackageManager and compose an application with just nexTemplateManager.

Since

1.7.0

4.54.2 Member Function Documentation

4.54.2.1 boolean com.nexstreaming.nexeditorsdk.nexTemplateManager.applyTemplateToProjectById (nexProject *project,* String *templateId*)

This applies a Template to a nexProject.

This method automatically applies effects to clips in a nexProject.

Parameters

project	The project to which Template will be applied to; there must be at least one video clip.
templateId	nexAssetPackageManager.ItemEnt.id() or nexAssetPackageManager.Item.id() of Template Category.

Returns

TRUE - Template is properly applied to the FALSE - Applying Template failed.

Since

1.7.0

4.54.2.2 int com.nexstreaming.nexeditorsdk.nexTemplateManager.findNewPackages ( )

This checks if there is any new Template package.

This API is same as nexAssetPackageManager#uninstallPackageById(String).

Returns

The number of new packages to install.

Since

1.7.0

4.54.2.3 String com.nexstreaming.nexeditorsdk.nexTemplateManager.getLastErrorMessage ( )

If applyTemplateToProjectById(nexProject, String) failed, this gets the error message which contains the cause.

Returns

`NULL` - OK; others - a message about exceptions that occurred.

Since

1.7.0

4.54.2.4 String [ ] com.nexstreaming.nexeditorsdk.nexTemplateManager.getTemplateAssetIds ( )

This gets the Asset Package IDs of the Templates. loadTemplate() has to be run at least once to get normal values.

Returns

See Also

loadTemplate()

Since

1.7.0

4.54.2.5 int [ ] com.nexstreaming.nexeditorsdk.nexTemplateManager.getTemplateAssetIdxs ( )

This gets the package assetIdx of the installed Templates.

Returns

The package assetIdx of the installed Templates.

Since

1.7.0

4.54.2.6 Template com.nexstreaming.nexeditorsdk.nexTemplateManager.getTemplateById (String *id*)

This gets the detailed information of the Template to the Template ID.

Parameters

id	id - nexAssetPackageManager.ItemEnt.id() or nexAssetPackageManager.Item.id() of Template Category.

Returns

Template information.

4.54.2.7 String [ ] com.nexstreaming.nexeditorsdk.nexTemplateManager.getTemplateIds ( )

This gets the IDs of the installed Templates.

Returns

The IDs of the installed Templates.

Since

1.7.0

4.54.2.8 static nexTemplateManager com.nexstreaming.nexeditorsdk.nexTemplateManager.getTemplateManager (Context *appContext*, Context *resContext*) [static]

Creates and gets the instance of nexTemplateManager. Call this API first to create an instance of nexTemplateManager.

Instances created by this API are saved as a `static` and shared with all.

Parameters

appContext	The application context.
resContext	The context with Template resources.

Returns

The singleton instance of nexTemplateManager.

See Also

getTemplateManager()

Since

1.5.42

4.54.2.9 static nexTemplateManager com.nexstreaming.nexeditorsdk.nexTemplateManager.getTemplateManager ( ) [static]

Gets the instances of nexTemplateManager that have been created.

Note

Do not call this API first. For it to work properly, getTemplateManager(Context, Context) should have been executed.

Returns

The singleton instance of nexTemplateManager. If getTemplateManager(Context, Context) was never called before, the return value will be `NULL`.

See Also

getTemplateManager(Context, Context)

Since

1.5.42

4.54.2.10 List<Template> com.nexstreaming.nexeditorsdk.nexTemplateManager.getTemplates ( )

This gets the template information being managed in the form of TemplateEntry.

Returns

The TemplateEntry of the templates.

Since

1.5.42

4.54.2.11 void com.nexstreaming.nexeditorsdk.nexTemplateManager.installPackagesAsync (final nexAssetPackageManager.OnInstallPackageListener *listener*)

This API installs Templates received from the Asset Store.

As a pre-requisite, run findNewPackages() first. If findNewPackages() finds a new package, then the developer should use it. installPackagesAsync is a single task, so check if it is running using isInstallingPackages() before using it. This API is same as nexAssetPackageManager#installPackagesAsync(nexAssetPackageManager.OnInstallPackageListener).

Parameters

listener	

Since

1.7.0

4.54.2.12 boolean com.nexstreaming.nexeditorsdk.nexTemplateManager.isInstallingPackages ( )

This API checks if installPackagesAsync is running or not.

This API is same as nexAssetPackageManager#isInstallingPackages().

Returns

Since

1.7.0

4.54.2.13 void com.nexstreaming.nexeditorsdk.nexTemplateManager.loadTemplate ( )

Loads the template resources from the assets and external drive.

This API works through blocking. For faster loading, it does not check template effects.

Since

1.5.42

4.54.2.14 void com.nexstreaming.nexeditorsdk.nexTemplateManager.uninstallPackageById (String *templateId*)

This uninstalls a Template.

This API is same as nexAssetPackageManager.uninstallPackageById(String).

Parameters

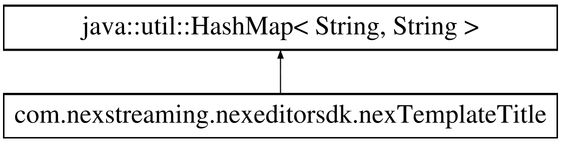
templateId	nexAssetPackageManager.ItemEnt.id() or nexAssetPackageManager.Item.id() of Template Category.

Since

1.7.0

4.55 com.nexstreaming.nexeditorsdk.nexTemplateTitle Class Reference

Inheritance diagram for com.nexstreaming.nexeditorsdk.nexTemplateTitle:


java::util::HashMap< String, String >
com.nexstreaming.nexeditorsdk.nexTemplateTitle

Public Member Functions

- nexTemplateTitle ()
- void release ()
- int **getTemplateEffectID** ()
- void **setTemplateEffectID** (int mTemplateEffectID)
- int **getTemplateSet** ()
- void **setTemplateSet** (int mTemplateSet)

4.55.1 Constructor & Destructor Documentation

4.55.1.1 com.nexstreaming.nexeditorsdk.nexTemplateTitle.nexTemplateTitle ( )

This method creates an instance of `nexTemplateTitle`.

Since

version 1.5.48

4.55.2 Member Function Documentation

4.55.2.1 void com.nexstreaming.nexeditorsdk.nexTemplateTitle.release ( )

This method clears the resources of `nexTemplateTitle`. This method must be called when the activity is destroyed.

Since

version 1.5.48

4.56 com.nexstreaming.nexeditorsdk.nexTheme Class Reference

Public Member Functions

- String getId ()

- String getName (Context context)
- String getDesc (Context context)
- Bitmap getIconSyncEx ()
- String getTransitionId ()
- String getOpeningTitleEffectId ()
- String getEndingTitleEffectId ()
- String getMiddleTitleEffectId ()
- String getAccentEffectId ()

Static Public Member Functions

- static boolean isValidThemeId (String Id)

4.56.1 Detailed Description

This class contains effects, for basic video editing, sorted in themes.

Example code:

class sample code

```
nexEffectLibrary fxlib = nexEffectLibrary.getEffectLibrary(this);
nexTheme[] allThemes = fxlib.getThemes();
for( nexTheme t: allThemes ) {
    Log.i(LOG_TAG,"Name =" + t.getName(getBaseContext()));
}
```

Since

version 1.0.0

4.56.2 Member Function Documentation

4.56.2.1 String com.nexstreaming.nexeditorsdk.nexTheme.getAccentEffectId ( )

This method gets the accent effect ID of a theme. The effect ID retrieved with this method can be used to make a separate clip effect.

Example code :

```
accenteffectid.setText(mAllThemes[index].getAccentEffectId());
```

Returns

The ID of an accent effect.

Since

version 1.3.3

4.56.2.2 String com.nexstreaming.nexeditorsdk.nexTheme.getDesc (Context *context*)

This method gets the description of a theme.

Example code :

```
desc.setText(mAllThemes[index].getDesc(getApplicationContext()));
```

Parameters

context	The Android context.

Returns

Description of the theme as a `string`.

Since

version 1.0.0

4.56.2.3 String com.nexstreaming.nexeditorsdk.nexTheme.getEndingTitleEffectId ( )

This method gets the ending effect ID of a theme.

The effect ID retrieved with this method can be used to make a separate clip effect.

Example code :

```
endtitle.setText(mAllThemes[index].getEndingTitleEffectId());
```

Returns

The ID of an ending effect.

Since

version 1.3.3

4.56.2.4 Bitmap com.nexstreaming.nexeditorsdk.nexTheme.getIconSyncEx ( )

This method gets the icon of a theme, as a bitmap.

Example code :

```
if( mAllThemes[index].getIconSyncEx() != null ){
    iv.setVisibility(View.VISIBLE);
    iv.setImageBitmap(mAllThemes[index].getIconSyncEx());
}
```

Returns

Bitmap of the theme icon.

Since

version 1.0.0

4.56.2.5 String com.nexstreaming.nexeditorsdk.nexTheme.getId ( )

This method gets the ID of a theme.

Example code :

```
id.setText(mAllThemes[index].getId());
```

Returns

Theme ID as a `String`.

Since

version 1.0.0

4.56.2.6 String com.nexstreaming.nexeditorsdk.nexTheme.getMiddleTitleEffectId ( )

This method gets the middle effect ID of a theme. Middle effect allows to set titles to the clip. The effect ID retrieved with this method can be used to make a separate clip effect.

Example code :

```
middletitle.setText(mAllThemes[index].getMiddleTitleEffectId());
```

Returns

The ID of a title effect.

Since

version 1.3.3

4.56.2.7 String com.nexstreaming.nexeditorsdk.nexTheme.getName (Context *context*)

This method gets the name of a theme.

Example code :

```
name.setText(mAllThemes[index].getName(mContext));
```

Parameters

context	The Android context.

Returns

A theme name as a `String`.

Since

version 1.0.0

4.56.2.8 String com.nexstreaming.nexeditorsdk.nexTheme.getOpeningTitleEffectId ( )

This method gets the opening effect ID of a theme.

The effect ID retrieved with this method can be used to make a separate clip effect.

Example code :

```
opentitle.setText(mAllThemes[index].getOpeningTitleEffectId());
```

Returns

The ID of an opening effect.

Since

version 1.3.3

4.56.2.9 String com.nexstreaming.nexeditorsdk.nexTheme.getTransitionId ( )

This method gets the transition effect ID of a theme.

Example code :

```
trid.setText(mAllThemes[index].getTransitionId());
```

Returns

The ID of the transition effect.

Since

version 1.3.3

4.56.2.10 static boolean com.nexstreaming.nexeditorsdk.nexTheme.isValidThemeId (String *Id*) [static]

This method checks whether an ID is a valid theme ID.

Example code :

```
if( nexTheme.isValidThemeId(ThemeId) ) {
    mThemeId = ThemeId;
}
```

Parameters

Id	The theme ID to check, as a String.

Returns

TRUE if the theme ID is usable for nexEffectLibrary, otherwise FALSE.

Since

version 1.0.0

4.57 com.nexstreaming.nexeditorsdk.nexOverlayManager.nexTitleInfo Class Reference

Public Member Functions

- **nexTitleInfo** (int id, int textLen, String textDesc)
- int getId ()
- String getText ()
- void setText (String text)
- String getTextDesc ()
- int getTextMaxLen ()

4.57.1 Detailed Description

This class is used to retrieve the title information of an overlay title template.

Since

version 1.7.0

4.57.2 Member Function Documentation

4.57.2.1 int com.nexstreaming.nexeditorsdk.nexOverlayManager.nexTitleInfo.getId ( )

This method is used to get the ID of title information.

Returns

Id of title information.

Since

version 1.7.0

4.57.2.2 String com.nexstreaming.nexeditorsdk.nexOverlayManager.nexTitleInfo.getText ( )

This method is used to get the title of title information.

Returns

Title of title information.

Since

version 1.7.0

4.57.2.3 String com.nexstreaming.nexeditorsdk.nexOverlayManager.nexTitleInfo.getTextDesc ( )

This method is used to get the text description of title information.

Returns

Description of title information.

Since

version 1.7.0

4.57.2.4 int com.nexstreaming.nexeditorsdk.nexOverlayManager.nexTitleInfo.getTextMaxLen ( )

This method is used to get the max length of input title.

Returns

Max length of input title.

Since

version 1.7.0

4.57.2.5 void com.nexstreaming.nexeditorsdk.nexOverlayManager.nexTitleInfo.setText (String *text*)

This method is used to set a new title string.

Parameters

text	New title string.

Since

version 1.7.0

4.58 com.nexstreaming.nexeditorsdk.nexTranscode Class Reference

Classes

- enum Error
- class OnTransCoderListener
- class Option
- enum Rotate
- enum State

Public Member Functions

- nexTranscode setTransCoderListener (OnTransCoderListener listener)
- nexTranscode (String sourcePath, boolean useMediaInfo)
- nexTranscode run (Option opt)
- boolean cancel ()
- int getProgress ()
- State getCurrentState ()
- Error getLastError ()

Static Public Member Functions

- static void init (nexEngine engine)
- static nexTranscode getTranscode (String sourcePath, OnTransCoderListener listener)

4.58.1 Detailed Description

This method is used when changing the bitrate and resolution of one file. Any work related to transcoding uses the device's hardware resources. Therefore, editing, previewing, or exporting will interrupt transcoding. Also, the same rule applies to making nexClips and extracting thumbnails from existing nexClips.

Example code :

```
 mEngine.stop();
nexTranscode.init(mEngine);
 mTransCoder = new nexTranscode("/sdard/input.mp4",false);
 mTransCoder.setTransCoderListener(new nexTranscode.OnTransCoderListener() {

     public void onProgress(int progress, int maxProgress) {
         //TODO: UI progress
     }

     public void onTransCodeDone(nexTranscode.Error error, int suberror) {
         //TODO: Done
     }
 });
 nexTranscode.Option opt = new nexTranscode.Option(new File("/sdard/output.mp4"), outputWidth, outputHeight
      , outputBitrate, true);
 mTransCoder.run(opt);
```

Since

version 1.5.29

4.58.2 Constructor & Destructor Documentation

4.58.2.1 com.nexstreaming.nexeditorsdk.nexTranscode.nexTranscode (String *sourcePath,* boolean *useMediaInfo*)

A constructor.

Parameters

sourcePath	The source content path.
useMediaInfo	-**TRUE**: Checks the content first. -**FALSE**: Does not check.

Since

version 1.5.29

4.58.3 Member Function Documentation

4.58.3.1 boolean com.nexstreaming.nexeditorsdk.nexTranscode.cancel ( )

run(Option) stops transcoding which was running.

Returns

`TRUE` if the transcoding is successful; otherwise, `FALSE`.

Since

version 1.5.29

4.58.3.2 State com.nexstreaming.nexeditorsdk.nexTranscode.getCurrentState ( )

This method gets the current state.

Returns

State

Since

version 1.5.29

4.58.3.3 Error com.nexstreaming.nexeditorsdk.nexTranscode.getLastError ( )

This method gets the last error value.

Returns

Error

Since

version 1.5.29

4.58.3.4 int com.nexstreaming.nexeditorsdk.nexTranscode.getProgress ( )

This method shows the transcoding progress in percentages.

Returns

Transcoding The progress.

Since

version 1.5.29

4.58.3.5 static nexTranscode com.nexstreaming.nexeditorsdk.nexTranscode.getTranscode (String *sourcePath,* OnTransCoderListener *listener*) [static]

This method gets the nexTranscode instance through the source path after setting the listener. In the future version, a newly created nexTranscode instance from this method will be managed separately.

Parameters

sourcePath	The source content path.
listener	The UI listener.

Returns

A newly created nexTranscode instance.

Since

version 1.5.29

4.58.3.6 static void com.nexstreaming.nexeditorsdk.nexTranscode.init (nexEngine *engine*) [static]

This method sets the engine needed for transcoding. Any work related to transcoding uses the engine: the user must call this method first. If there are no further changes to the engine, calling is unnecessary.

Parameters

engine	The engine instance used in the editor.

Since

version 1.5.29

4.58.3.7 nexTranscode com.nexstreaming.nexeditorsdk.nexTranscode.run (Option *opt*)

Executes transcoding. Calling this API will start transcoding via nexEngine, which means nexEngine must be in the 'stop' status.

Parameters

opt	The option for transcoding.

Returns

this

Since

version 1.5.29

4.58.3.8 nexTranscode com.nexstreaming.nexeditorsdk.nexTranscode.setTransCoderListener (OnTransCoderListener *listener*)

This method sets the transcoding listener.

Parameters

listener	The UI listener.

Returns

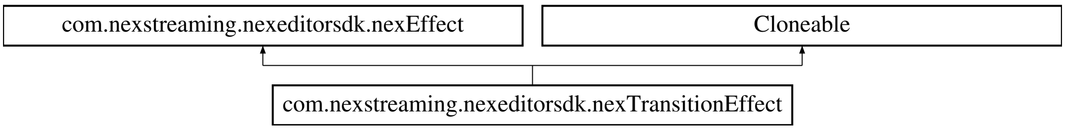
this nexTranscode.

Since

version 1.5.29

4.59 com.nexstreaming.nexeditorsdk.nexTransitionEffect Class Reference

Inheritance diagram for com.nexstreaming.nexeditorsdk.nexTransitionEffect:


com.nexstreaming.nexeditorsdk.nexEffect
Cloneable
com.nexstreaming.nexeditorsdk.nexTransitionEffect

Public Member Functions

- void setTransitionEffect (String ID)
- void setAutoTheme ()
- int getMinDuration ()
- int getMaxDuration ()
- int getOffset ()
- int getOverlap ()
- String getName (Context context)
- String getDesc (Context context)
- Bitmap getIcon ()
- String getCategoryTitle (Context context)

Static Protected Member Functions

- static nexTransitionEffect **clone** (nexTransitionEffect src)

Additional Inherited Members

4.59.1 Detailed Description

This class sets transition effects to be used by the NexEditor™ SDK. With transition effects, the user can set various effects to switch from one clip to the next.

Example code 1:

class sample code

```
nexEffectLibrary fxlib = nexEffectLibrary.getEffectLibrary(this);
nexTransitionEffect[] transitionEffects = fxlib.getTransitionEffects();
for (nexTransitionEffect e : transitionEffects) {
    mEffectItem.add(e.getId());
}
```

Example code 2:

class sample code

```
project.getClip(0,true).getTransitionEffect().setTransitionEffect(effectID);
```

See Also

nexClip::getTransitionEffect()
findTransitionEffectById
nexEffectLibrary::getTransitionEffects()

Since

version 1.0.0

4.59.2 Member Function Documentation

4.59.2.1 String com.nexstreaming.nexeditorsdk.nexTransitionEffect.getCategoryTitle (Context *context*)

This method gets the category name of a transition effect.

Example code :

```
name.setText(mTransitionEffects[index].getName(mContext)+":"+mTransitionEffects[index].
      getCategoryTitle(mContext));
```

Parameters

context	The Android context.

Returns

The category name as a `String`.

Since

version 1.3.3

4.59.2.2 String com.nexstreaming.nexeditorsdk.nexTransitionEffect.getDesc (Context *context*)

This method gets the description of a transition effect.

Example code :

```
desc.setText(mTransitionEffects[index].getDesc(mContext));
```

Parameters

context	The Android context.

Returns

Returns one of:

- Description of the effect if it is not included in any existing theme,
- "None Transition" if there is no transition effect set between the clips, or
- "Theme Transition" if the transition effect that is set between the clips is part of an existing theme.

Since

version 1.0.0

4.59.2.3 Bitmap com.nexstreaming.nexeditorsdk.nexTransitionEffect.getIcon ( )

This method gets the icon of a transition effect.

Example code :

```
if( mTransitionEffects[index].getIcon() != null ){
    iv.setVisibility(View.VISIBLE);
    iv.setImageBitmap(mTransitionEffects[index].getIcon());
}
```

Returns

An instance of the transition effect icon as a bitmap.

Since

version 1.0.0

4.59.2.4 int com.nexstreaming.nexeditorsdk.nexTransitionEffect.getMaxDuration ( )

This method gets the maximum duration for a particular transition effect.

Example code :

```
int maxDuration = project.getClip(0,true).getTransitionEffect().getMaxDuration();
```

Returns

The maximum duration of the transition effect in `msec` (milliseconds).

Since

version 1.0.0

See Also

getMinDuration()

4.59.2.5 int com.nexstreaming.nexeditorsdk.nexTransitionEffect.getMinDuration ( )

This method gets the minimum duration for a particular transition effect.

Example code :

```
int minDuration = project.getClip(0,true).getTransitionEffect().getMinDuration();
```

Returns

The minimum duration of the transition effect in `msec` (milliseconds).

Since

version 1.0.0

See Also

getMaxDuration()

4.59.2.6 String com.nexstreaming.nexeditorsdk.nexTransitionEffect.getName (Context *context*)

This method gets the name of a transition effect.

Example code :

```
name.setText(mTransitionEffects[index].getName(mContext)+":"+mTransitionEffects[index].
      getCategoryTitle(mContext));
```

Parameters

context	The Android context.

Returns

Returns one of:

- The name of the transition effect if it is not part of any existing theme,
- "None Transition" if there is no transition effect set between the clips, or
- "Theme Transition" if the transition effect that is set between the clips is part of an existing theme.

Since

version 1.0.0

4.59.2.7 int com.nexstreaming.nexeditorsdk.nexTransitionEffect.getOffset ( )

This method gets the beginning position of a transition effect, relative to the end of the first clip in the transition.

The value returned is a percentage of the total duration of the transition effect. This means that if an example transition effect has a total duration of 4 seconds, and begins 1 second before the end of the first clip in a transition, then the value returned here would be 25 (percent).

The offset depends on the duration of time that neighboring clips overlap during the transition effect, so this value depends on the overlap percentage that can be retrieved by calling the method `getOverlap()`.

Example code :

```
visualClip.mEffectOffset = clipItem.getTransitionEffect().getOffset();
```

Returns

The offset of the beginning of the transition effect from the end of the first clip in the transition, as a percentage of the total duration of the transition effect.

See Also

getOverlap()

Since

version 1.0.0

4.59.2.8 int com.nexstreaming.nexeditorsdk.nexTransitionEffect.getOverlap ( )

This method gets the duration of time that clips overlap during a transition effect, as a percentage of the total transition time.

The overlap is indicated as a percentage of the total duration of the transition effect, so for example, if a transition effect takes 4 seconds to complete, and the neighboring clips overlap for 2 seconds of that transition, the value returned here for the example transition effect will be 50.

Example code :

```
visualClip.mEffectOverlap = clipItem.getTransitionEffect().getOverlap();
```

Returns

The duration of time that clips overlap during the transition effect as a percentage of the total time of the transition effect, as an `integer`.

See Also

getOffset()

Since

version 1.0.0

4.59.2.9 void com.nexstreaming.nexeditorsdk.nexTransitionEffect.setAutoTheme ( )

This method sets a transition effect on a clip matching the theme currently set.

Example code :

```
project.getClip(0,true).getTransitionEffect().setAutoTheme();
```

Since

version 1.0.0

See Also

setTransitionEffect(String)
setEffectNone()

4.59.2.10 void com.nexstreaming.nexeditorsdk.nexTransitionEffect.setTransitionEffect (String *ID*)

This method sets a transition effect, that is not part of any existing theme, on a clip.

Example code :

```
clip.getTransitionEffect().setTransitionEffect("com.nexstreaming.unkownid");
```

Parameters

ID	The transition effect ID as a `String`.

Exceptions

com.nexstreaming.nexeditorsdk.exception.InvalidEffectIDException	

See Also

setAutoTheme()
setEffectNone()

Since

version 1.0.0

See Also

getId()

4.60 com.nexstreaming.nexeditorsdk.nexUtils Class Reference

Static Public Member Functions

- static double noise (double xin, double yin, double zin)

4.60.1 Detailed Description

NexEditor™ provides methods that are difficult to implement in the application.

4.60.2 Member Function Documentation

4.60.2.1 static double com.nexstreaming.nexeditorsdk.nexUtils.noise (double *xin,* double *yin,* double *zin*) [static]

Use this method to make an animation that moves to random coordinates in nexAnimate.

Parameters

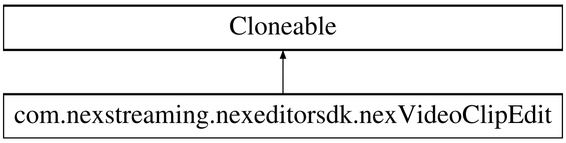
xin	The x coordinate.
yin	The y coordinate.
zin	The z coordinate or the time value.

Returns

The value to which the noise has been applied.

4.61 com.nexstreaming.nexeditorsdk.nexVideoClipEdit Class Reference

Inheritance diagram for com.nexstreaming.nexeditorsdk.nexVideoClipEdit:



Public Member Functions

- void setTrim (int startTime, int endTime)
- int getStartTrimTime ()
- int getEndTrimTime ()
- void addTrim (int startTime, int endTime, int speed)
- int removeTrim (int loc)
- int getTrimCount ()
- void setSpeedControl (int speed)
- int getSpeedControl ()
- int setAutoTrim (int trimMode, int value)
- void clearTrim ()
- int getDuration ()

Static Public Attributes

- static int kAutoTrim_Divided = 1
- static int kAutoTrim_Interval = 2
- static final int kSpeedControl_MinValue = 12
- static final int kSpeedControl_MaxValue = 400

Static Protected Member Functions

- static nexVideoClipEdit **clone** (nexVideoClipEdit src)

4.61.1 Detailed Description

This class handles playback speed and trim options of a video clip in the NexEditor™ .

Example code 1:

```
mEngine.getProject().getClip(0,true).getVideoClipEdit().setSpeedControl(200);
mEngine.play();
```

Example code 2:

```
if( project.getTotalClipCount(true) == 1 ){
    nexClip clip = project.getClip(0,true);
    if( clip.getClipType() == nexClip.kCLIP_TYPE_VIDEO ){
        if(clip.getTotalTime() > 12000 ) {
            clip.getVideoClipEdit().setAutoTrim(nexVideoClipEdit.kAutoTrim_Divided, 3);
        }
    }
}
```

Since

version 1.0.0

4.61.2 Member Function Documentation

4.61.2.1 void com.nexstreaming.nexeditorsdk.nexVideoClipEdit.addTrim (int *startTime,* int *endTime,* int *speed*)

Deprecated This method can be used to trim several parts of a video clip in different ways.

Trimmed video clips can be set to different playback speeds, which ignore the overall video playback speed set by setSpeedControl(int).

Parameters

startTime	The start time of a video clip in `msec` (milliseconds).
endTime	The end time of a video clip in `msec` (milliseconds).
speed	The playback speed of the trimmed clip as a percentage of the original speed (from 25 (1/4 the original speed) up to 200 (twice the original speed)).

See Also

removeTrim(int)

Since

version 1.0.0

Deprecated For internal use only. Please do not use.

4.61.2.2 void com.nexstreaming.nexeditorsdk.nexVideoClipEdit.clearTrim ( )

This method clears the trim information set to a clip. When trim information is cleared from the clip, the clip will return to its original state.

Example code :

```
clip.getVideoClipEdit().clearTrim();
```

See Also

setTrim(int, int)

Since

version 1.3.43

4.61.2.3 int com.nexstreaming.nexeditorsdk.nexVideoClipEdit.getDuration ( )

This method gets the duration of a video clip within a project after being edited. For example, the video clip could be trimmed or speed adjusted so that the duration is different from the original duration.

Example code :

```
int duration = clip.getVideoClipEdit().getDuration();
```

Returns

The duration of the video clip within a project in `msec` (milliseconds).

Since

version 1.3.43

4.61.2.4 int com.nexstreaming.nexeditorsdk.nexVideoClipEdit.getEndTrimTime ( )

This method gets the end time of the trimmed section of a clip.

A clip can be trimmed into a desired section by adjusting both ends (beginning and ending) of the clip. The end time of this method is based on the total play time of the clip.

Example code :

```
trim = mPrimaryItems.get(postion).getVideoClipEdit().getEndTrimTime();
```

Returns

The end time of the trimmed section, in `msec` (milliseconds), as an `integer`.

See Also

getStartTrimTime()

Since

version 1.1.0

4.61.2.5 int com.nexstreaming.nexeditorsdk.nexVideoClipEdit.getSpeedControl ( )

This method gets the playback speed set for a video clip.

Example code :

```
int speedcontrol = mVideoEdit.getSpeedControl();
```

Returns

The playback speed of the video clip as a percentage of the original speed (100).

See Also

setSpeedControl(int)

Since

version 1.0.0

4.61.2.6 int com.nexstreaming.nexeditorsdk.nexVideoClipEdit.getStartTrimTime ( )

This method gets the start time of the trimmed section of a clip.

A clip can be trimmed into a desired section by adjusting both ends (beginning and ending) of the clip. The start time of this method is based on the total play time of the clip.

Example code :

```
trim = mPrimaryItems.get(postion).getVideoClipEdit().getStartTrimTime();
```

Returns

The start time of the trimmed section, in `msec` (milliseconds), as an `integer`.

See Also

getEndTrimTime()

Since

version 1.1.0

4.61.2.7 int com.nexstreaming.nexeditorsdk.nexVideoClipEdit.getTrimCount ( )

Deprecated This method gets the number of times trim has been set with the method addTrim() within a specific clip.

Returns

The number of trims set for the specific clip.

Since

version 1.0.0

See Also

addTrim(int, int, int)
removeTrim(int)

For internal use only. Please do not use.

4.61.2.8 int com.nexstreaming.nexeditorsdk.nexVideoClipEdit.removeTrim (int *loc*)

Deprecated This method removes a trim set on a video clip.

Parameters

loc	The location of the trim to be removed.

Since

version 1.0.0

See Also

addTrim(int, int, int)
setTrim(int, int)

For internal use only. Please do not use.

4.61.2.9 int com.nexstreaming.nexeditorsdk.nexVideoClipEdit.setAutoTrim (int *trimMode,* int *value*)

Deprecated This method trims a video clip automatically, based on the mode chosen with the parameter, `trimMode`.

Parameters

trimMode	How to automatically trim video clips. This should be one of:

- kAutoTrim_Divided = { kAutoTrim_Divided}
- kAutoTrim_Interval = { kAutoTrim_Interval}

Parameters

value	Additional information on how the clip should be trimmed based on the value of `trimMode`. • When trimMode = `kAutoTrim_Divided`, this value is the number of equally sized clips that should be created (Min 2 ∼ Max 10). The duration of each clip should not be less than 4 seconds, otherwise an error code will be returned. • When trimMode = `kAutoTrim_Interval`, this value is the interval time in milliseconds that thumbnail images will be extracted from the original clip.(Min 1000)

Returns

0 if successful, -1 if the value of the parameter `value` is not in the range or -2 if the value is not applicable on the current video clip.

See Also

getTrimCount()

Since

version 1.0.0

Deprecated For internal use only. Please do not use.

4.61.2.10 void com.nexstreaming.nexeditorsdk.nexVideoClipEdit.setSpeedControl (int *speed*)

This method sets the playback speed of a video clip. The value set by this method will be ignored if the video clip was trimmed at any time.

Example code :

```
clip.getVideoClipEdit().setSpeedControl(200);
```

Parameters

speed	The new playback speed of the video clip as a percentage of the original speed. ({ kSpeedControl_MinValue}% (slow) ∼ 100%(standard) ∼ { kSpeedControl_MaxValue}%(fast)

See Also

getSpeedControl()

Since

version 1.0.0

4.61.2.11 void com.nexstreaming.nexeditorsdk.nexVideoClipEdit.setTrim (int *startTime,* int *endTime*)

This method clears previous trim history and starts a new trim.

Example code :

```
mProject.getClip(0, true).getVideoClipEdit().setTrim(0, trim);
```

Parameters

startTime	The start time of the video clip in `msec` (milliseconds).
endTime	The end time of the video clip in `msec` (milliseconds).

See Also

addTrim(int, int, int)

Exceptions

com.nexstreaming.nexeditorsdk.exception.InvalidRangeException	

Since

version 1.0.0

4.61.3 Member Data Documentation

4.61.3.1 int com.nexstreaming.nexeditorsdk.nexVideoClipEdit.kAutoTrim_Divided = 1 `[static]`

One of the possible values of the first parameter, trimMode, of the method setAutoTrim(int, int). If trimMode is set to this value, the clip will be divided equally into the number of videos set by the second parameter, value. For example, if value = 5, the original video clip will be equally divided into 5 shorter clips.

Since

version 1.0.0

See Also

setAutoTrim(int, int)

Deprecated For internal use only. Please do not use.

4.61.3.2 int com.nexstreaming.nexeditorsdk.nexVideoClipEdit.kAutoTrim_Interval = 2 `[static]`

One of the possible values of the first parameter, trimMode, of the method setAutoTrim(int, int). If trimMode is set to this value, the second parameter, value, is the interval time in `milliseconds` used to save thumbnail images from the original clip at determined intervals.

Since

version 1.0.0

See Also

setAutoTrim(int, int)

Deprecated For internal use only. Please do not use.

4.61.3.3 final int com.nexstreaming.nexeditorsdk.nexVideoClipEdit.kSpeedControl_MaxValue = 400 [static]

The possible maximum speed, as a percentage or the original speed, to set as the parameter `speed` of the method `setSpeedControl()`.

Since

version 1.3.0

See Also

setSpeedControl(int)

4.61.3.4 final int com.nexstreaming.nexeditorsdk.nexVideoClipEdit.kSpeedControl_MinValue = 12 [static]

The possible minimum speed, as a percentage or the original speed, to set as the parameter `speed` of the method `setSpeedControl()`.

Since

version 1.3.0

See Also

setSpeedControl(int)

4.62 com.nexstreaming.nexeditorsdk.nexEngine.OnAutoTrimResultListener Class Reference

Public Member Functions

- abstract void onAutoTrimResult (int[] arrayData)

4.62.1 Detailed Description

This abstract class manages the `AutoTrim` results received from the background.

The AutoTrim is a feature supported by the NexEditor™ that returns requested number of trimmed sections from one original video content. The returned value is in array of start timestamps which are the start time position of each of the trimmed sections.

Since

version 1.5.15

4.62.2 Member Function Documentation

4.62.2.1 abstract void com.nexstreaming.nexeditorsdk.nexEngine.OnAutoTrimResultListener.onAutoTrimResult (int[] *arrayData*) [abstract]

This method receives the `AutoTrim` result sent from the background.

Parameters

arrayData	An array of requested start timestamps.

Since

version 1.5.15

4.63 com.nexstreaming.nexeditorsdk.nexEngine.OnCompletionListener Class Reference

Public Member Functions

- abstract void **onComplete** (int resultCode)

4.63.1 Detailed Description

This abstract class handles the listener used when `stop()` is called.

Since

version 1.0.0

See Also

stop()
#exportSaveStop(OnCompletionListener)

4.64 com.nexstreaming.nexeditorsdk.nexApplicationConfig.OnEffectLoadListener Class Reference

Public Member Functions

- abstract void onEffectPriorityLoadComplete ()
- abstract void onEffectLoadComplete ()
- abstract void onEffectLoad (String packageDesc, int error)

4.64.1 Detailed Description

A listener class of setEffectLoadListener(). For more details, please refer to setEffectLoadListener(OnEffectLoadListener).

See Also

setEffectLoadListener(OnEffectLoadListener)

Since

1.5.42

4.64.2 Member Function Documentation

4.64.2.1 abstract void com.nexstreaming.nexeditorsdk.nexApplicationConfig.OnEffectLoadListener.onEffectLoad (String *packageDesc,* int *error*) [abstract]

Returns the name of the KMT package that just finished loading.

Parameters

packageDesc	The name of the KMT package that has loaded.
error	The error value from loading. Zero for success, or a non-zero NexEditor™ error code in the event of a failure.

Since

1.5.42

4.64.2.2 abstract void com.nexstreaming.nexeditorsdk.nexApplicationConfig.OnEffectLoadListener.onEffectLoadComplete ( ) [abstract]

Called when loading all effects has finished.

Since

1.5.42

4.64.2.3 abstract void com.nexstreaming.nexeditorsdk.nexApplicationConfig.OnEffectLoadListener.onEffectPriorityLoadComplete ( ) [abstract]

Called when loading a high-priority effect has finished.

Since

1.5.42

4.65 com.nexstreaming.nexeditorsdk.nexClip.OnGetVideoClipDetailThumbnailsListener Class Reference

Public Member Functions

- abstract void onGetDetailThumbnailResult (int event, Bitmap bm, int index, int totalCount, int timestamp)

Static Public Attributes

- static int **kEvent_Ok** = 0
- static int **kEvent_Completed** = 1
- static int **kEvent_Fail** = -1
- static int **kEvent_systemError** = -2
- static int **kEvent_UserCancel** = -3

4.65.1 Detailed Description

This class is a listener used with onGetDetailThumbnailResult(int, android.graphics.Bitmap, int, int, int).

See Also

onGetDetailThumbnailResult(int, android.graphics.Bitmap, int, int, int)

Since

version 1.3.0

4.65.2 Member Function Documentation

4.65.2.1 abstract void com.nexstreaming.nexeditorsdk.nexClip.OnGetVideoClipDetailThumbnailsListener.onGetDetailThumbnailResult (int *event*, Bitmap *bm*, int *index*, int *totalCount*, int *timestamp*) [abstract]

This method returns whether retrieving process of `detailThumbnail` was successful or not.

Parameters

event	A `kEvents` of the class `OnGetVideoClipDetailThumbnailsListener` to tell the result of `detailThumbnail` retrieving.
bm	Retrieved thumbnails in a bitmap format.
index	The index of retrieved thumbnail as `integer`.
totalCount	The total number of retrieved thumbnails as `integer`.
timestamp	The timestamps of each of retrieved thumbnails as `integer`.

Since

version 1.3.0

4.66 com.nexstreaming.nexeditorsdk.nexClip.OnGetVideoClipIDR2YOnlyThumbnailsListener Class Reference

Public Member Functions

- abstract void onGetVideoClipIDR2YOnlyThumbnailsResult (int event, byte[] data, int index, int totalCount, int timestamp)

Static Public Attributes

- static int kEvent_Ok = 0
- static int kEvent_Completed = 1
- static int kEvent_Fail = -1
- static int kEvent_systemError = -2
- static int **kEvent_UserCancel** = -3

4.66.1 Detailed Description

Deprecated For internal use only. Please do not use.

Since

version 1.3.43

4.66.2 Member Function Documentation

4.66.2.1 abstract void com.nexstreaming.nexeditorsdk.nexClip.OnGetVideoClipIDR2YOnlyThumbnailsListener.onGetVideoClipIDR2YOnlyThumbnailsResult (int *event*, byte[ ] *data*, int *index*, int *totalCount*, int *timestamp*) [abstract]

This method gets the details of a retrieved thumbnail.

This method is called asynchronously by onGetVideoClipIDR2YOnlyThumbnailsResult(int, byte[], int, int, int).

Parameters

event	The kEvents of the class `OnGetVideoClipDetailThumbnailsListener`.
data	The instance of the thumbnail bitmap returned from the `nexEngine`. This instance operates in singleton so it should be saved separately to avoid overwrite with other bitmaps.
index	The index of the current thumbnail from the total thumbnails retrieved.
totalCount	The total number of parsed thumbnails.
timestamp	The timestamp of the thumbnails returned from the `nexEngine`.

Since

version 1.3.0

4.66.3 Member Data Documentation

4.66.3.1 int com.nexstreaming.nexeditorsdk.nexClip.OnGetVideoClipIDR2YOnlyThumbnailsListener.kEvent_Completed = 1 [static]

This event indicates that all the thumbnail retrieving operation has been completed.

Since

version 1.3.0

4.66.3.2 int com.nexstreaming.nexeditorsdk.nexClip.OnGetVideoClipIDR2YOnlyThumbnailsListener.kEvent_Fail = -1 [static]

This event indicates that the thumbnail retrieving was unsuccessful.

Since

version 1.3.0

4.66.3.3 int com.nexstreaming.nexeditorsdk.nexClip.OnGetVideoClipIDR2YOnlyThumbnailsListener.kEvent_Ok = 0 [static]

This event indicates that the thumbnail was successfully retrieved.

Since

version 1.3.0

4.66.3.4 int com.nexstreaming.nexeditorsdk.nexClip.OnGetVideoClipIDR2YOnlyThumbnailsListener.kEvent_systemError = -2 [static]

This event indicates that a new thumbnail task couldn't be generated.

Since

version 1.3.0

4.67 com.nexstreaming.nexeditorsdk.nexAssetPackageManager.OnInstallPackageListener Class Reference

Public Member Functions

- abstract void onProgress (int countPackage, int totalPackages, int progressInstalling)
- abstract void onCompleted (int event)

Static Public Attributes

- static int kEvent_installOk = 0
- static int kEvent_IinstallFail = -1

4.67.1 Detailed Description

This is the listener used in installPackagesAsync.

See Also

#installPackagesAsync(OnInstallPackageListener)

Since

1.7.0

4.67.2 Member Function Documentation

4.67.2.1 abstract void com.nexstreaming.nexeditorsdk.nexAssetPackageManager.OnInstallPackageListener.onCompleted (int *event*) [abstract]

This notifies the end of all installations.

Parameters

event	kEvent_installOk - OK, kEvent_linstallFail - fail

Since

1.7.0

4.67.2.2 abstract void com.nexstreaming.nexeditorsdk.nexAssetPackageManager.OnInstallPackageListener.onProgress (int *countPackage,* int *totalPackages,* int *progressInstalling*) [abstract]

This notifies the installation progress of installPackagesAsync.

Parameters

countPackage	Count of packages currently installing. (1 ∼ totalPackages)
totalPackages	Count of new packages found using nexAssetPackageManager#findNewPackages().
progressInstalling	The installation progress of packages currently installing. (0 ∼ 100)

Since

1.7.0

4.67.3 Member Data Documentation

4.67.3.1 int com.nexstreaming.nexeditorsdk.nexAssetPackageManager.OnInstallPackageListener.kEvent_installOk = 0 [static]

Since

1.7.0

4.67.3.2 int com.nexstreaming.nexeditorsdk.nexAssetPackageManager.OnInstallPackageListener.kEvent_linstallFail = -1 [static]

Since

1.7.0

4.68 com.nexstreaming.nexeditorsdk.nexEffectLibrary.OnInstallPluginEffectPackage-AsyncListener Class Reference

Public Member Functions

- abstract void **onPackageLoaded** (int discount)
- abstract void **onComplete** (int failedCount, String[] errorMessages)

4.69 com.nexstreaming.nexeditorsdk.nexClip.OnLoadVideoClipThumbnailListener Class Reference

Public Member Functions

- abstract void onLoadThumbnailResult (int event)

Static Public Attributes

- static int kEvent_Ok = 0
- static int kEvent_loadCompleted = 1
- static int kEvent_Running = 2
- static int kEvent_mustRetry = 3
- static int kEvent_loadFail = -1
- static int kEvent_systemError = -2

4.69.1 Detailed Description

This class operates asynchronously and defines the events sent when `loadVideoClipThumbnails()` takes too much time getting thumbnails from a clip. The event, loadVideoClipThumbnails(com.nexstreaming.nexeditorsdk.nexClip.OnLoadVideoClipThumbnailListener), indicates when a thumbnail has been loaded successfully.

See Also

loadVideoClipThumbnails(com.nexstreaming.nexeditorsdk.nexClip.OnLoadVideoClipThumbnailListener)

Since

version 1.0.0

4.69.2 Member Function Documentation

4.69.2.1 abstract void com.nexstreaming.nexeditorsdk.nexClip.OnLoadVideoClipThumbnailListener.onLoadThumbnailResult (int *event*) `[abstract]`

This event is called when `loadVideoClipThumbnails()` operation is complete.

Parameters

event	Set to 0 if works normally, otherwise set to -1.

See Also

loadVideoClipThumbnails(com.nexstreaming.nexeditorsdk.nexClip.OnLoadVideoClipThumbnailListener)

Since

version 1.0.0

4.69.3 Member Data Documentation

4.69.3.1 int com.nexstreaming.nexeditorsdk.nexClip.OnLoadVideoClipThumbnailListener.kEvent_loadCompleted = 1 [static]

This event indicates that the thumbnail has already retrieved.

Since

version 1.3.0

4.69.3.2 int com.nexstreaming.nexeditorsdk.nexClip.OnLoadVideoClipThumbnailListener.kEvent_loadFail = -1 [static]

This event indicates that thumbnail retrieving has failed.

Since

version 1.3.0

4.69.3.3 int com.nexstreaming.nexeditorsdk.nexClip.OnLoadVideoClipThumbnailListener.kEvent_mustRetry = 3 [static]

This event indicates that the thumbnail retrieving operation was unsuccessful due to other engine operation. Try again later.

Since

version 1.3.0

4.69.3.4 int com.nexstreaming.nexeditorsdk.nexClip.OnLoadVideoClipThumbnailListener.kEvent_Ok = 0 [static]

This event indicates that the thumbnail was successfully retrieved.

Since

version 1.3.0

4.69.3.5 int com.nexstreaming.nexeditorsdk.nexClip.OnLoadVideoClipThumbnailListener.kEvent_Running = 2 [static]

This event indicates that the thumbnail is currently being retrieved.

Since

version 1.3.0

4.69.3.6 int com.nexstreaming.nexeditorsdk.nexClip.OnLoadVideoClipThumbnailListener.kEvent_systemError = -2 [static]

This event indicates that a new thumbnail task couldn't be generated.

Since

version 1.3.0

4.70 com.nexstreaming.nexeditorsdk.nexEngine.OnSeekCompletionListener Class Reference

Public Member Functions

- abstract void **onSeekComplete** (int resultCode, int seekTime, int movedTime)

4.70.1 Detailed Description

This abstract class handles the listener used when `seekIDROnly()` is called.

See Also

#seekIDROnly(int, OnSeekCompletionListener)

Since

version 1.3.34

4.71 com.nexstreaming.nexeditorsdk.nexEngine.OnSurfaceChangeListener Class Reference

Public Member Functions

- abstract void **onSurfaceChanged** ()

4.71.1 Detailed Description

This class is a listener for `setOnSurfaceChangeListener`.

Since

version 1.0.0

See Also

setOnSurfaceChangeListener(OnSurfaceChangeListener)

4.72 com.nexstreaming.nexeditorsdk.nexTranscode.OnTransCoderListener Class Reference

Public Member Functions

- abstract void **onProgress** (int progress, int maxProgress)
- abstract void **onTransCodeDone** (Error error, int suberror)

4.72.1 Detailed Description

The listener of nexTranscode.

See Also

setTransCoderListener(OnTransCoderListener)

Since

version 1.5.29

4.73 com.nexstreaming.nexeditorsdk.nexTranscode.Option Class Reference

Public Member Functions

- Option (File outputFile, int outputWidth, int outputHeight, int outputBitRate, int outputSamplingRate, boolean outputFit)
- Option (File outputFile, int outputWidth, int outputHeight, int outputBitRate, int outputSamplingRate, boolean outputFit, nexTranscode.Rotate outputRotateMeta, nexTranscode.Rotate outputRotate)
- void **setOutputRotateMeta** (nexTranscode.Rotate value)
- void **setOutputRotate** (nexTranscode.Rotate value)

4.73.1 Detailed Description

The output option of nexTranscode.

See Also

run(Option)

Since

version 1.5.29

4.73.2 Constructor & Destructor Documentation

4.73.2.1 com.nexstreaming.nexeditorsdk.nexTranscode.Option.Option (File *outputFile,* int *outputWidth,* int *outputHeight,* int *outputBitRate,* int *outputSamplingRate,* boolean *outputFit*)

The constructor of the output option in nexTranscode.

Parameters

outputFile	The output file. The user must create directories in advance.
outputWidth	The width of the output.
outputHeight	The height of the output.
outputBitRate	The bitrate of the output.
outputFit	Set to `TRUE` to create black bars(letterbox or pillarbox) to prevent any distortions in the original. Set to `FALSE` to distort the original to fit to the output resolution.

Since

version 1.5.29

4.73.2.2 com.nexstreaming.nexeditorsdk.nexTranscode.Option.Option (File *outputFile,* int *outputWidth,* int *outputHeight,* int *outputBitRate,* int *outputSamplingRate,* boolean *outputFit,* nexTranscode.Rotate *outputRotateMeta,* nexTranscode.Rotate *outputRotate*)

The constructor of the output option in nexTranscode.

Parameters

outputFile	The output file. The user must create directories in advance.
outputWidth	The width of the output.
outputHeight	The height of the output.
outputBitRate	The bitrate of the output.
outputFit	Set to `TRUE` to create black bars(letterbox or pillarbox) to prevent any distortions in the original. Set to `FALSE` to distort the original to fit to the output resolution.
outputRotateMeta	The metaData rotation value(BYPASS,CW_0,CW_90,CW_180,CW_270)

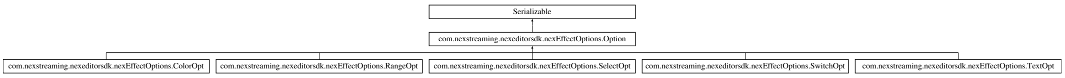
outputRotate	The image rotation value(clockwise standard:CW_0,CW_90,CW_180,CW_270)

Since

version 1.5.35

4.74 com.nexstreaming.nexeditorsdk.nexEffectOptions.Option Class Reference

Inheritance diagram for com.nexstreaming.nexeditorsdk.nexEffectOptions.Option:


Serializable
com.nexstreaming.nexeditorsdk.nexEffectOptions.Option
com.nexstreaming.nexeditorsdk.nexEffectOptions.ColorOpt
com.nexstreaming.nexeditorsdk.nexEffectOptions.RangeOpt
com.nexstreaming.nexeditorsdk.nexEffectOptions.SelectOpt
com.nexstreaming.nexeditorsdk.nexEffectOptions.SwitchOpt
com.nexstreaming.nexeditorsdk.nexEffectOptions.TextOpt

Public Member Functions

- String getId ()
- String getLabel ()

Protected Member Functions

- **Option** (String id, String label)

4.74.1 Detailed Description

This class defines and saves the possible text, color, and select options for an effect.

The effect options, `Options`, defined by this class create a serialization, which allows the values to be passed to other applications or activities.

Since

version 1.0.1

4.74.2 Member Function Documentation

4.74.2.1 String com.nexstreaming.nexeditorsdk.nexEffectOptions.Option.getId ( )

This method gets the saved ID of an effect option.

Returns

The ID of the effect option, as a `String`.

Since

version 1.0.1

4.74.2.2 String com.nexstreaming.nexeditorsdk.nexEffectOptions.Option.getLabel ( )

This method gets the label of an effect option.

Based on the label returned, it is possible to guess which option is in use.

Returns

The option label, as a `String`.

Since

version 1.0.1

4.75 com.nexstreaming.nexeditorsdk.nexOverlayManager.Overlay Class Reference

Public Member Functions

- String getId ()
- String getName (String locale)
- Bitmap getIcon ()

4.75.1 Detailed Description

This class is used to retrieve the overlay title template.

Since

version 1.7.0

4.75.2 Member Function Documentation

4.75.2.1 Bitmap com.nexstreaming.nexeditorsdk.nexOverlayManager.Overlay.getIcon ( )

This method is used to get the bitmap icon of an overlay title template.

Returns

Bitmap icon of an overlay title template.

Since

version 1.7.0

4.75.2.2 String com.nexstreaming.nexeditorsdk.nexOverlayManager.Overlay.getId ( )

This method is used to get the ID of an overlay title template.

Returns

ID of an overlay title template.

Since

version 1.7.0

4.75.2.3 String com.nexstreaming.nexeditorsdk.nexOverlayManager.Overlay.getName (String *locale*)

This method is used to get the name of an overlay title template.

Parameters

locale	Encoded locale of name.

Returns

Name of an overlay title template.

Since

version 1.7.0

4.76 com.nexstreaming.nexeditorsdk.nexEngine.OverlayPreviewBuilder Class Reference

Public Member Functions

- OverlayPreviewBuilder setOutline (boolean show)
- OverlayPreviewBuilder setPositionX (int pos)
- OverlayPreviewBuilder setPositionY (int pos)
- OverlayPreviewBuilder setRotateX (int degree)
- OverlayPreviewBuilder setRotateY (int degree)
- OverlayPreviewBuilder setRotateZ (int degree)
- OverlayPreviewBuilder setScaleX (float ratio)
- OverlayPreviewBuilder setScaleY (float ratio)
- OverlayPreviewBuilder setAlpha (float value)
- void display ()
- void clear ()

4.76.1 Detailed Description

This class builds an overlay item preview.

Since

version 1.5.19

See Also

buildOverlayPreview(int)

4.76.2 Member Function Documentation

4.76.2.1 void com.nexstreaming.nexeditorsdk.nexEngine.OverlayPreviewBuilder.clear ( )

This method clears all the overlay items.

Example code :

```
mEngine.buildOverlayPreview(mOverlayId).clear();
```

Since

version 1.5.15

4.76.2.2 void com.nexstreaming.nexeditorsdk.nexEngine.OverlayPreviewBuilder.display ( )

This method displays an overlay item.

Example code :

```
mEngine.buildOverlayPreview(mOverlayId).setOutline(false);
    mEngine.buildOverlayPreview(mOverlayId).setOutline(true).setScaleY(scale);
    mEngine.buildOverlayPreview(mOverlayId).setOutline(true).setAlpha(lastProgress);
    mEngine.buildOverlayPreview(mOverlayId).display();
```

Since

version 1.5.15

4.76.2.3 OverlayPreviewBuilder com.nexstreaming.nexeditorsdk.nexEngine.OverlayPreviewBuilder.setAlpha (float *value*)

This method sets the transparency of an overlay item.

Example code :

```
mEngine.buildOverlayPreview(mOverlayId).setOutline(true).setAlpha(lastProgress).
      display();
```

Parameters

value	The transparency value.

Returns

OverlayPreviewBuilder

Since

version 1.5.15

4.76.2.4 OverlayPreviewBuilder com.nexstreaming.nexeditorsdk.nexEngine.OverlayPreviewBuilder.setOutline (boolean *show*)

This method decides whether to show the outline of an overlay item.

Example code :

```
mEngine.buildOverlayPreview(mOverlayId).setOutline(false).display();
```

Parameters

show	Set to `true` to show outline.

Returns

OverlayPreviewBuilder

Since

version 1.5.15

4.76.2.5 OverlayPreviewBuilder com.nexstreaming.nexeditorsdk.nexEngine.OverlayPreviewBuilder.setPositionX (int *pos*)

This method sets the X-axis of an overlay item.

Example code :

```
mEngine.buildOverlayPreview(mOverlayId).setOutline(true).setPositionX((int) (event.getX() * 1280
      / v.getWidth()))
```

Parameters

pos	The X-axis of the overlay item.

Returns

OverlayPreviewBuilder

Since

version 1.5.15

4.76.2.6 OverlayPreviewBuilder com.nexstreaming.nexeditorsdk.nexEngine.OverlayPreviewBuilder.setPositionY (int *pos*)

This method sets the Y-axis of an overlay item.

Example code :

```
mEngine.buildOverlayPreview(mOverlayId).setOutline(true).setPositionY((int) (event.getY() * 720 /
       v.getHeight()))
```

Parameters

pos	The Y-axis of the overlay item.

Returns

OverlayPreviewBuilder

Since

version 1.5.15

4.76.2.7 OverlayPreviewBuilder com.nexstreaming.nexeditorsdk.nexEngine.OverlayPreviewBuilder.setRotateX (int *degree*)

This method sets the degree value to rotate an overlay item around its X-axis.

Example code :

```
mEngine.buildOverlayPreview(mOverlayId).setOutline(true).setRotateX(angle).
      display();
```

Parameters

degree	The degree value to rotate.

Returns

OverlayPreviewBuilder

Since

version 1.5.15

4.76.2.8 OverlayPreviewBuilder com.nexstreaming.nexeditorsdk.nexEngine.OverlayPreviewBuilder.setRotateY (int *degree*)

This method sets the degree value to rotate an overlay item around its Y-axis.

Example code :

```
mEngine.buildOverlayPreview(mOverlayId).setOutline(true).setRotateY(angle).
      display();
```

Parameters

degree	The degree value to rotate.

Returns

OverlayPreviewBuilder

Since

version 1.5.15

4.76.2.9 OverlayPreviewBuilder com.nexstreaming.nexeditorsdk.nexEngine.OverlayPreviewBuilder.setRotateZ (int *degree*)

This method sets the degree value to rotate an overlay item around its Z-axis.

Example code :

```
mEngine.buildOverlayPreview(mOverlayId).setOutline(true).setRotateZ(angle).
      display();
```

Parameters

degree	The degree value to rotate.

Returns

OverlayPreviewBuilder

Since

version 1.5.15

4.76.2.10 OverlayPreviewBuilder com.nexstreaming.nexeditorsdk.nexEngine.OverlayPreviewBuilder.setScaleX (float *ratio*)

This method sets the scale value to scale an overlay item along the X-axis.

Example code :

```
mEngine.buildOverlayPreview(mOverlayId).setOutline(true).setScaleX(scale).
      display();
```

Parameters

ratio	The degree value to rotate.

Returns

OverlayPreviewBuilder

Since

version 1.5.15

4.76.2.11 OverlayPreviewBuilder com.nexstreaming.nexeditorsdk.nexEngine.OverlayPreviewBuilder.setScaleY (float *ratio*)

This method sets the scale value to scale an overlay item along the Y-axis.

Example code :

```
mEngine.buildOverlayPreview(mOverlayId).setOutline(true).setScaleY(scale).
      display();
```

Parameters

ratio	The degree value to rotate.

Returns

OverlayPreviewBuilder

Since

version 1.5.15

4.77 com.nexstreaming.nexeditorsdk.nexAssetPackageManager.PreAssetCategoryAlias Enum Reference

Public Attributes

- **Effect**
- **Transition**
- **Font**
- **Overlay**
- **Audio**
- **Template**
- **ClipGraphics**
- **TextEffect**
- **Extra**

4.77.1 Detailed Description

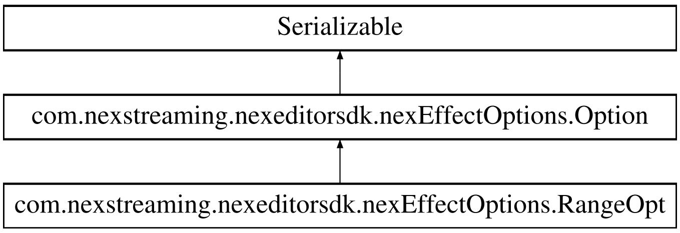
A category of an Asset Package. This is different from the category of item in an Asset. This category is used in the VAsset Store Application, and the values are used for when an Asset Package wants to find out the directory of the category. The number of categories may increase as the SDK version number goes up.

See Also

Asset#getCategoryAlias()
RemoteAssetInfo#getCategoryAlias()

4.78 com.nexstreaming.nexeditorsdk.nexEffectOptions.RangeOpt Class Reference

Inheritance diagram for com.nexstreaming.nexeditorsdk.nexEffectOptions.RangeOpt:


Serializable
com.nexstreaming.nexeditorsdk.nexEffectOptions.Option
com.nexstreaming.nexeditorsdk.nexEffectOptions.RangeOpt

Public Member Functions

- void setValue (int value)
- int getValue ()
- int min ()
- int max ()

Protected Member Functions

- **RangeOpt** (String id, String label, int default_value, int min_value, int max_value)

Protected Attributes

- int **default_value**
- int **min_value**
- int **max_value**
- int **mValue**

4.78.1 Detailed Description

This is an option that receives input similar to a slider.

Since

1.7.0

4.78.2 Member Function Documentation

4.78.2.1 int com.nexstreaming.nexeditorsdk.nexEffectOptions.RangeOpt.getValue ( )

Returns the currently set value.

Returns

The set value.

Since

1.7.0

4.78.2.2 int com.nexstreaming.nexeditorsdk.nexEffectOptions.RangeOpt.max ( )

Returns the maximum value that can be set.

Returns

max value

Since

1.7.0

4.78.2.3 int com.nexstreaming.nexeditorsdk.nexEffectOptions.RangeOpt.min ( )

Returns the minimum value that can be set.

Returns

min value

Since

1.7.0

4.78.2.4 void com.nexstreaming.nexeditorsdk.nexEffectOptions.RangeOpt.setValue (int *value*)

Input setting values. Value should be between the min and max values. If the number is smaller than min(), it will be substitued by min() value; vice versa.

Parameters

value	min() ∼ max(),

Since

1.7.0

4.79 com.nexstreaming.nexeditorsdk.nexAssetPackageManager.RemoteAssetInfo Interface Reference

Inherited by com.nexstreaming.nexeditorsdk.nexAssetPackageManager.RemoteAssetInfoEnt.

Public Member Functions

- int idx ()
- String id ()
- String name ()
- Bitmap icon ()
- String getCategoryAlias ()

4.79.1 Detailed Description

This is a class that notifies the information of a featured Asset in the Asset Store.

This is not an installed Asset, but a link of a featured Asset in the Asset Store.

See Also

getRemoteAssetInfos(int)
getRemoteAssetInfo(int)

Since

1.7.0

4.79.2 Member Function Documentation

4.79.2.1 String com.nexstreaming.nexeditorsdk.nexAssetPackageManager.RemoteAssetInfo.getCategoryAlias ( )

The Asset Category alias name.

Returns

Since

1.7.1

4.79.2.2 Bitmap com.nexstreaming.nexeditorsdk.nexAssetPackageManager.RemoteAssetInfo.icon ( )

The Asset icon.

Returns

Since

1.7.0

4.79.2.3 String com.nexstreaming.nexeditorsdk.nexAssetPackageManager.RemoteAssetInfo.id ( )

The Asset ID.

Returns

Since

1.7.0

4.79.2.4 int com.nexstreaming.nexeditorsdk.nexAssetPackageManager.RemoteAssetInfo.idx ( )

The Asset idx. Page redirection is available with this value.

Returns

Since

1.7.0

4.79.2.5 String com.nexstreaming.nexeditorsdk.nexAssetPackageManager.RemoteAssetInfo.name ( )

The Asset Name.

Returns

Since

1.7.0

4.80 com.nexstreaming.nexeditorsdk.nexTranscode.Rotate Enum Reference

Public Attributes

- **BYPASS**
- **CW_0**
- **CW_90**
- **CW_180**
- **CW_270**

4.80.1 Detailed Description

The value of Rotation.

See Also

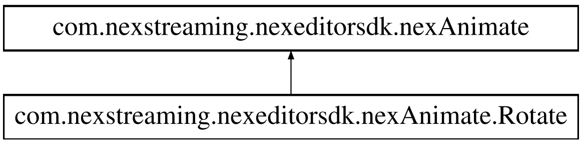
#Option(File outputFile, int outputWidth, int outputHeight, int outputBitRate, boolean outputFit, nexTranscode.Rotate outputRotateMeta, nexTranscode.Rotate outputRotate)

Since

vesion 1.5.35

4.81 com.nexstreaming.nexeditorsdk.nexAnimate.Rotate Class Reference

Inheritance diagram for com.nexstreaming.nexeditorsdk.nexAnimate.Rotate:


com.nexstreaming.nexeditorsdk.nexAnimate
com.nexstreaming.nexeditorsdk.nexAnimate.Rotate

Protected Member Functions

- **Rotate** (int startTime, int duration, boolean clockwise, int rotateDegree, int axis, MoveTrackingPath center)
- float **getAngleDegree** (int time, float startDegree, int axis)
- float **getTranslatePosition** (int currTime, int axis)

Additional Inherited Members

4.82 com.nexstreaming.nexeditorsdk.nexOverlayImage.runTimeMakeBitMap Interface Reference

Public Member Functions

- boolean isAniMate ()
- int getBitmapID ()
- Bitmap makeBitmap ()

4.82.1 Detailed Description

This interface allows the developer to make bitmap in runtime. This interface is used by nexOverlayImage(String, #runTimeMakeBitMap) as parameter `runTimeMakeBitMap`.

The following sample code demonstrates internal process :

Example code :

```
 if( isAniMate() ){
for( ; ; ){
getBitmapID();
makeBitmap()
}
} else {
getBitmapID();
makeBitmap()
}
```

See Also

#nexOverlayImage(String, com.nexstreaming.nexeditorsdk.nexOverlayImage.runTimeMakeBitMap)

Since

version 1.3.0

4.82.2 Member Function Documentation

4.82.2.1 int com.nexstreaming.nexeditorsdk.nexOverlayImage.runTimeMakeBitMap.getBitmapID ( )

This method gets the bitmap ID of an image to be used by the method `makeBitmap()`.

The developer should assign IDs to every bitmaps and the IDs should not be repeated with other image IDs or android resource IDs. When making bitmaps for an animation, every bitmaps of the animation should have different IDs. Every ID should be a number bigger than 2.

Returns

The ID of a bitmap to be used by `makeBitmap()`.

Since

version 1.3.3

4.82.2.2 boolean com.nexstreaming.nexeditorsdk.nexOverlayImage.runTimeMakeBitMap.isAniMate ( )

This method returns whether the runtime bitmap is for an animation or a still image by the return value, to the `nexEngine`.

If returned value is `FALSE`, `getBitmapID()` and `makeBitmap()` will be called only once.

Returns

`TRUE` if the bitmap is for an animation; `FALSE` if the bitmap is for a still image.

Since

version 1.3.3

4.82.2.3 Bitmap com.nexstreaming.nexeditorsdk.nexOverlayImage.runTimeMakeBitMap.makeBitmap ( )

This method makes the bitmap with the ID from the method `getBitmapID()` to make an overlay image.

Returns

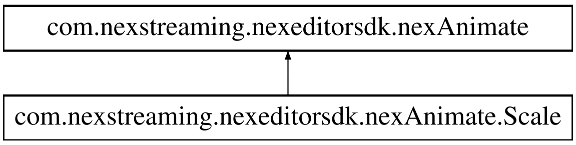
The bitmap to be used as an overly image.

Since

version 1.3.3

4.83 com.nexstreaming.nexeditorsdk.nexAnimate.Scale Class Reference

Inheritance diagram for com.nexstreaming.nexeditorsdk.nexAnimate.Scale:


com.nexstreaming.nexeditorsdk.nexAnimate
com.nexstreaming.nexeditorsdk.nexAnimate.Scale

Protected Member Functions

- **Scale** (int startTime, int duration, float lastScaledX, float lastScaledY, float lastScaledZ)
- **Scale** (int startTime, int duration, float startScaledX, float startScaledY, float startScaledZ, float lastScaledX, float lastScaledY, float lastScaledZ)
- **Scale** (int startTime, int duration, MoveTrackingPath scale)
- float **getScaledRatio** (int time, float startScaledRatio, int axis)

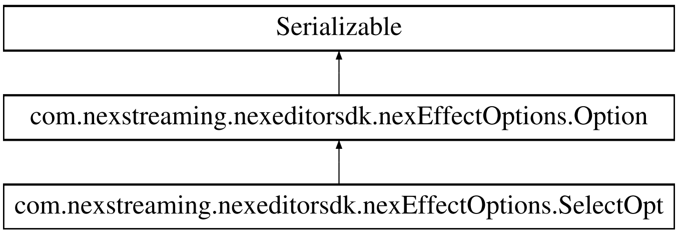
Protected Attributes

- final float **mLastScaledX**
- final float **mLastScaledY**
- final float **mLastScaledZ**
- final float **mStartScaledX**
- final float **mStartScaledY**
- final float **mStartScaledZ**
- final boolean **mSetStart**

Additional Inherited Members

4.84 com.nexstreaming.nexeditorsdk.nexEffectOptions.SelectOpt Class Reference

Inheritance diagram for com.nexstreaming.nexeditorsdk.nexEffectOptions.SelectOpt:


Serializable
com.nexstreaming.nexeditorsdk.nexEffectOptions.Option
com.nexstreaming.nexeditorsdk.nexEffectOptions.SelectOpt

Public Member Functions

- int getSelectIndex ()
- void setSelectIndex (int index)
- String[] getItems ()
- String getSelectValue ()

Protected Member Functions

- **SelectOpt** (String id, String label, String[] items, String[] values, int select)
- int **setValue** (String value)

Protected Attributes

- int **default_select_index**
- int **select_index**

4.84.1 Detailed Description

This class defines options that must be selected for an effect, such as text position and radio select options (allows the user to choose only one of a predefined set of options).

Example code:

```
for( nexEffectOptions.SelectOpt opt : select){
            if( f.getId().compareTo(opt.getId()) == 0 ){
                int val = opt.getSelectIndex();
                List<UserField.FieldOption> list = f.getOptions();
                b.append(URLEncoder.encode(opt.getId(), "UTF-8"));
                b.append("=");
                b.append(URLEncoder.encode(opt.getSelectValue(),"UTF-8"));
            }
        }
```

See Also

getSelectOptions()

Since

version 1.0.1

4.84.2 Member Function Documentation

4.84.2.1 String [ ] com.nexstreaming.nexeditorsdk.nexEffectOptions.SelectOpt.getItems ( )

This method gets items that can be selected for a particular select effect option, `SelectOpt`.

The sequence of the items determines their index number.

Returns

item The items that can be selected for the select effect option, as an array of `Strings`.

Since

version 1.0.1

4.84.2.2 int com.nexstreaming.nexeditorsdk.nexEffectOptions.SelectOpt.getSelectIndex ( )

This method gets the option index number of the currently selected option.

Returns

The index number of the selected option, as an `integer`.

Since

version 1.0.1

4.84.2.3 String com.nexstreaming.nexeditorsdk.nexEffectOptions.SelectOpt.getSelectValue ( )

This method gets the value selected for a particular select effect option, `SelectOpt`.

Returns

The selected value of the option, as a `String`.

Since

version 1.1.0

4.84.2.4 void com.nexstreaming.nexeditorsdk.nexEffectOptions.SelectOpt.setSelectIndex (int *index*)

This method sets the index number of the currently selected option item.

Parameters

index	The index number to set, as an `integer`.

Since

version 1.0.1

4.85 com.nexstreaming.nexeditorsdk.nexTranscode.State Enum Reference

Public Attributes

- **NONE**
- **IDLE**
- **WAIT**
- **RUNNING**
- **COMPLETE**

4.85.1 Detailed Description

The state of nexTranscode.

See Also

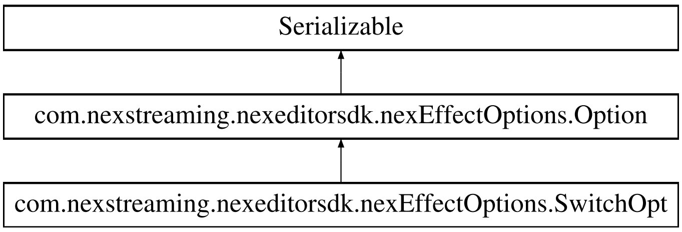
getCurrentState()

Since

version 1.5.29

4.86 com.nexstreaming.nexeditorsdk.nexEffectOptions.SwitchOpt Class Reference

Inheritance diagram for com.nexstreaming.nexeditorsdk.nexEffectOptions.SwitchOpt:


Serializable
com.nexstreaming.nexeditorsdk.nexEffectOptions.Option
com.nexstreaming.nexeditorsdk.nexEffectOptions.SwitchOpt

Public Member Functions

- boolean getValue ()
- void setValue (boolean on)

Protected Member Functions

- **SwitchOpt** (String id, String label, boolean on)

Protected Attributes

- boolean **default_on**
- boolean **on**

4.86.1 Detailed Description

An option that sets `ON/OFF`.

Since

1.7.0

4.86.2 Member Function Documentation

4.86.2.1 boolean com.nexstreaming.nexeditorsdk.nexEffectOptions.SwitchOpt.getValue ( )

This gets the set value.

Returns

`TRUE` - on, `FALSE` - off.

Since

1.7.0

4.86.2.2 void com.nexstreaming.nexeditorsdk.nexEffectOptions.SwitchOpt.setValue (boolean *on*)

This inputs values to set.

Parameters

on	`TRUE` - on , `FALSE` - off

Since

1.7.0

4.87 com.nexstreaming.nexeditorsdk.nexTemplateManager.Template Class Reference

Inherits com.nexstreaming.nexeditorsdk.nexAssetPackageManager.ItemEnt.

Public Member Functions

- String id ()
- String name (String locale)
- float aspect ()
- float[] getSupportedAspects ()
- void selectAspect (int index)
- String defaultBGMId ()

Static Public Member Functions

- static Template promote (nexAssetPackageManager.Item item)

4.87.1 Detailed Description

Item nexAssetPackageManager.Item interface of nexAssetPackageManager was inherited to make this class.

This processes Template Items in an AssetPackage as a group. Many Templates in an AssetPackage are of the same theme and different resolution, so the developer may choose a resolution via Template#selectAspect(int). The resolution set to the SDK is selected by default, but 16:89 is selected in case where there is no resolution supported.

Since

1.7.0

4.87.2 Member Function Documentation

4.87.2.1 float com.nexstreaming.nexeditorsdk.nexTemplateManager.Template.aspect ( )

The currently set Aspect ratio value.

Returns

16/9 - 16:9 , 1 - 1:1 , 9/16 - 9:16 , 2 - 2:1

Since

1.7.0

4.87.2.2 String com.nexstreaming.nexeditorsdk.nexTemplateManager.Template.defaultBGMId ( )

This gets the ID of the default BGM of a Template.

Returns

ID of the audio category of an Asset.

Since

1.7.1

4.87.2.3 float [ ] com.nexstreaming.nexeditorsdk.nexTemplateManager.Template.getSupportedAspects ( )

The supported Aspect ratio values.

Returns

16/9 - 16:9 , 1 - 1:1 , 9/16 - 9:16 , 2 - 2:1

Since

1.7.0

4.87.2.4 String com.nexstreaming.nexeditorsdk.nexTemplateManager.Template.id ( )

The Asset ID.

Returns

The Asset ID.

Since

1.7.0

Implements com.nexstreaming.nexeditorsdk.nexAssetPackageManager.Item.

4.87.2.5 String com.nexstreaming.nexeditorsdk.nexTemplateManager.Template.name (String *locale*)

This gets the item label. If there is no label set to the Asset, it is same as id().

Parameters

locale	The language or region information. For example, "en" - English , "ko_KR"- Korean, `NULL` is default value.

Returns

The label name.

Since

1.7.0

Implements com.nexstreaming.nexeditorsdk.nexAssetPackageManager.Item.

4.87.2.6 static Template com.nexstreaming.nexeditorsdk.nexTemplateManager.Template.promote (nexAssetPackageManager.Item *item*) [static]

If the category of com.nexstreaming.nexeditorsdk.nexAssetPackageManager.Item is Template, this API can create a Template.

Parameters

item	- Item received using nexAssetPackageManager#getInstalledAssetItemById(String)

Returns

- `NULL` if the Item category is not template.

Since

1.7.0

4.87.2.7 void com.nexstreaming.nexeditorsdk.nexTemplateManager.Template.selectAspect (int *index*)

The Aspect ratio index to set.

Parameters

index	Value smaller than the length of getSupportedAspects().

See Also

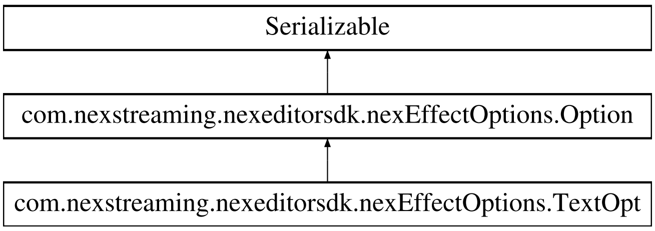
getSupportedAspects()

Since

1.7.0

4.88 com.nexstreaming.nexeditorsdk.nexEffectOptions.TextOpt Class Reference

Inheritance diagram for com.nexstreaming.nexeditorsdk.nexEffectOptions.TextOpt:


Serializable
com.nexstreaming.nexeditorsdk.nexEffectOptions.Option
com.nexstreaming.nexeditorsdk.nexEffectOptions.TextOpt

Public Member Functions

- void setText (String text)
- String getText ()

Protected Member Functions

- **TextOpt** (String id, String label, int lineNum)

4.88.1 Detailed Description

This class defines the text-related options possible in an effect.

Text-related options may include text position or if the the text in the effect appears on multiple lines among others.

Example code:

```
for( nexEffectOptions.TextOpt opt : text ) {
            if( f.getId().compareTo(opt.getId()) == 0 ){
                if( opt.getText() != null ){
                    b.append(URLEncoder.encode(opt.getId(), "UTF-8"));
                    b.append("=");
                    b.append(URLEncoder.encode(opt.getText(),"UTF-8"));
                }
            }
        }
```

See Also

getTextOptions()

Since

version 1.1.0

4.88.2 Member Function Documentation

4.88.2.1 String com.nexstreaming.nexeditorsdk.nexEffectOptions.TextOpt.getText ( )

This method gets the text saved for an effect.

Returns

The text saved on an effect, as a `String`.

See Also

setText(String)

Since

version 1.1.0

4.88.2.2 void com.nexstreaming.nexeditorsdk.nexEffectOptions.TextOpt.setText (String *text*)

This method sets the text input added to an effect.

Parameters

text	The text to be added to the effect, as a `String`.

See Also

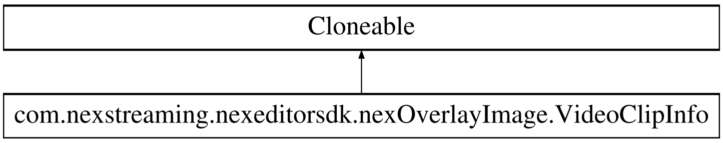
getText()

Since

version 1.1.0

4.89 com.nexstreaming.nexeditorsdk.nexOverlayImage.VideoClipInfo Class Reference

Inheritance diagram for com.nexstreaming.nexeditorsdk.nexOverlayImage.VideoClipInfo:



Public Member Functions

- boolean hasVideo ()
- boolean hasAudio ()
- String getPath ()
- int getTotalTime ()
- int getWidth ()
- int getHeight ()

Static Protected Member Functions

- static VideoClipInfo **clone** (VideoClipInfo src)

4.89.1 Detailed Description

The information of a video clip. The information includes details on the video content such as width, height, size and etc.

Example code :

```
 mVideoClipInfo = new VideoClipInfo();
mVideoClipInfo.mHasAudio = clip.hasAudio();
mVideoClipInfo.mHasVideo = clip.hasVideo();
mVideoClipInfo.mPath = clip.getPath();
mVideoClipInfo.mTotalTime = clip.getTotalTime();
mVideoClipInfo.mWidth = clip.getWidth();
mVideoClipInfo.mHeight = clip.getHeight();
```

Since

version 1.5.0

4.89.2 Member Function Documentation

4.89.2.1 int com.nexstreaming.nexeditorsdk.nexOverlayImage.VideoClipInfo.getHeight ( )

This method gets the height of a video clip.

Returns

The height of the video clip.

See Also

getWidth()

Since

version 1.5.0

4.89.2.2 String com.nexstreaming.nexeditorsdk.nexOverlayImage.VideoClipInfo.getPath ( )

This method gets the absolute path of the original video content of a video clip.

Returns

The absolute path of the video content.

Since

version 1.5.0

4.89.2.3 int com.nexstreaming.nexeditorsdk.nexOverlayImage.VideoClipInfo.getTotalTime ( )

This method gets the duration of a video clip.

Returns

The duration of the video clip.

Since

version 1.5.0

4.89.2.4 int com.nexstreaming.nexeditorsdk.nexOverlayImage.VideoClipInfo.getWidth ( )

This method gets the width of a video clip.

Returns

The width of the video clip.

See Also

getHeight()

Since

version 1.5.0

4.89.2.5 boolean com.nexstreaming.nexeditorsdk.nexOverlayImage.VideoClipInfo.hasAudio ( )

This method checks whether an audio content is assigned to a specific clip.

Returns

`TRUE` if an audio exists, otherwise `FALSE`.

See Also

hasVideo()

Since

version 1.5.0

4.89.2.6 boolean com.nexstreaming.nexeditorsdk.nexOverlayImage.VideoClipInfo.hasVideo ( )

This method checks whether a video content is assigned to a specific clip.

Returns

`TRUE` if a video exists, otherwise `FALSE`.

See Also

hasAudio()

Since

version 1.5.0

Index