

Demo Application UI Guide (Video Player)

Version 0.3.0

Display Audio

Solution Team



Release information

The following changes have been made to this document.

Change History

Date	Change
06 Dec. 2017	First release for v0.1.0
15 Dec. 2017	Update Storage Event v0.2.0
07 Feb. 2018	Update scenario v0.3.0

Proprietary Notice

Information in this document is provided solely to enable system and software implementers to use Nexell products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Nexell reserves the right to make changes without further notice to any products herein.

Nexell makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Nexell assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters which may be provided in Nexell data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals” must be validated for each customer application by customer’s technical experts. Nexell does not convey any license under its patent rights nor the rights of others. Nexell products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Nexell product could create a situation where personal injury or death may occur. Should Buyer purchase or use Nexell products for any such unintended or unauthorized application, Buyer shall indemnify and hold Nexell and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Nexell was negligent regarding the design or manufacture of the part.

Copyright© 2017 Nexell Co.,Ltd. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electric or mechanical, by photocopying, recording, or otherwise, without the prior written consent of Nexell.

Contact us

[11595] BundangYemiji Bldg. 12F, 31 Hwangsaeul-ro 258 beon gil, Bundang-gu, Sungnam-city, Gyeonggi-do, Korea.

TEL: 82-31-698-7400

FAX:82-31-698-7455

<http://www.nexell.co.kr>

Contents

Chap 1.	Overview	1
	1.1 Overview	1
	1.2 UI Guide.....	1
	1.2.1 Features	1
	1.2.2 Execute	1
	1.2.3 UI Details	2
	1.2.4 UI Playlist Details	2
	1.3 Flow.....	3
	1.3.1 Application Overall	3
	1.3.2 Application Switching	3
Chap 2.	GUI Guide	4
	2.1 Status Bar	4
	2.1.1 Home	4
	2.1.2 Back	4
	2.2 Media Player Control.....	5
	2.2.1 Initialization	5
	2.2.2 Volume Bar	5
	2.2.3 Progress Bar	5
	2.2.4 Prev Button	6
	2.2.5 Play Button	6
	2.2.6 Pause Button	7
	2.2.7 Next Button	7
	2.2.8 Stop Button	8
	2.2.9 Playlist Button	8
Chap 3.	Subtitle	10
	3.1 Overview	10
	3.2 Application Overall Flow	10
Chap 4.	Storage Event	11
	4.1 OverView	11
	4.2 Flow.....	11
	4.2.1 Removing Storage Event	11
	4.2.2 Inserting Storage Event	11
Chap 5.	Known Issues	12
	5.1 To Do List.....	12

Chap 1. Overview

1.1 Overview

This document describes how to use NxVideoPlayer application.

NxVideoPlayer makes video file list by using sqliteutils that reads database written by media scanner.

NxVideoPlayer also makes xml file that stores last playing information to play again from last state.

1.2 UI Guide

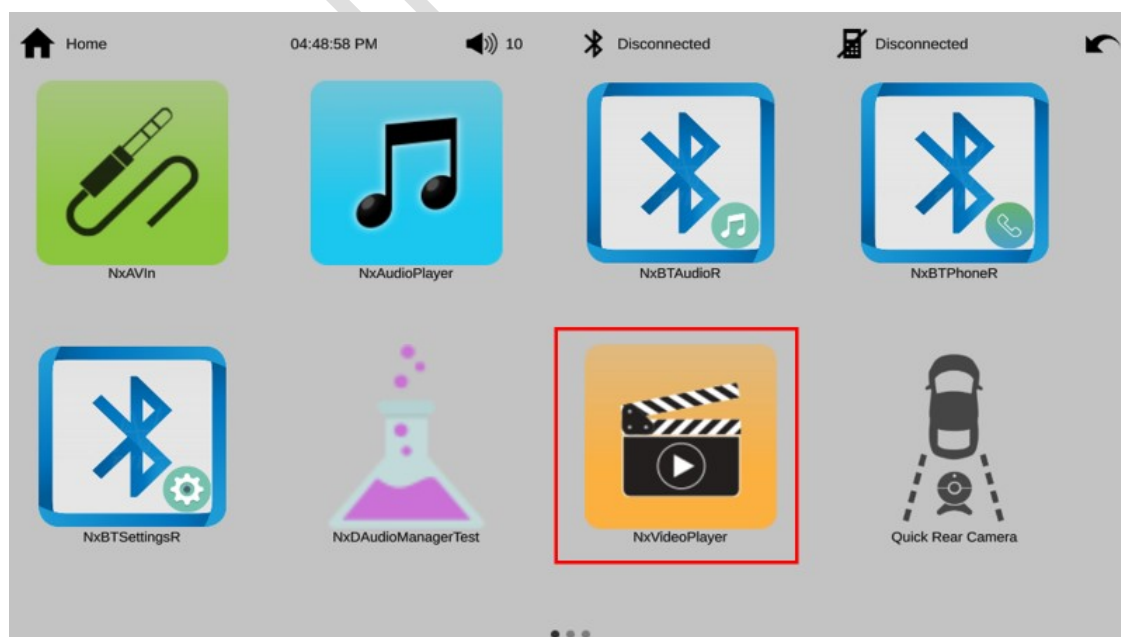
1.2.1 Features

- Video file play, seek, pause, and resume
- Make video file list including USB and SD-card.
- Auto play from last state, if available.
- Shows subtitle, if available.

1.2.2 Execute

NxVideoPlayer is executed by clicking icon in the launcher.

When some application is clicked, launcher hides from screen and clicked application appears.

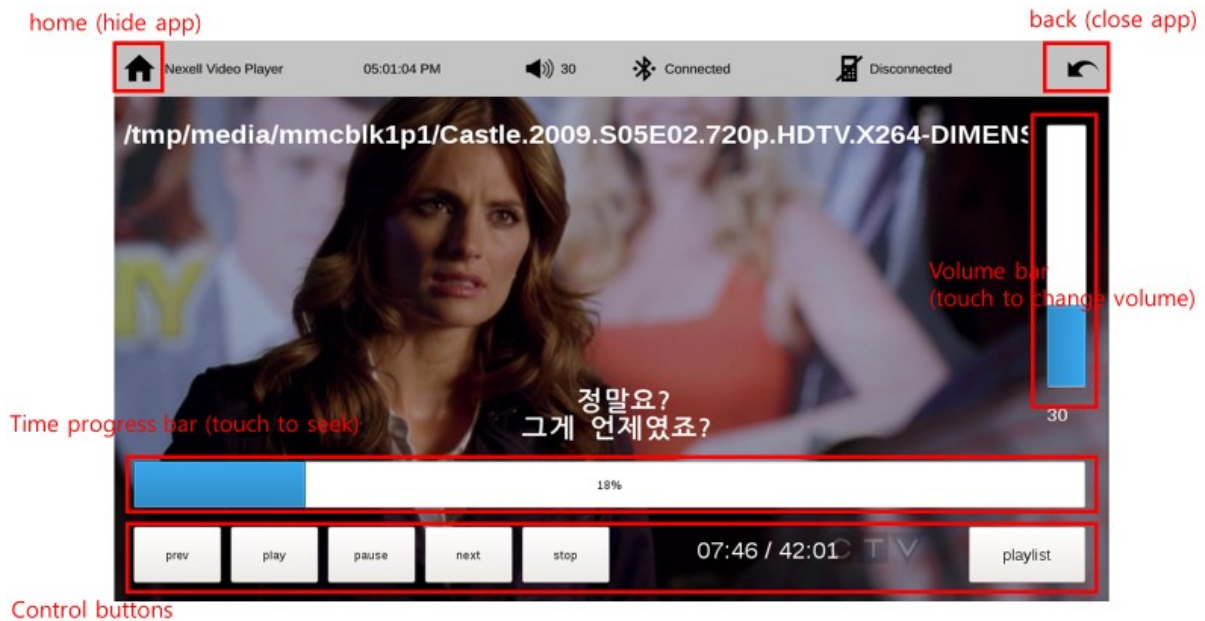


1.2.3 UI Details

GUI is shown like below picture after NxVideoPlayer is executed.

home (hide app) does not mean application window operates hide.

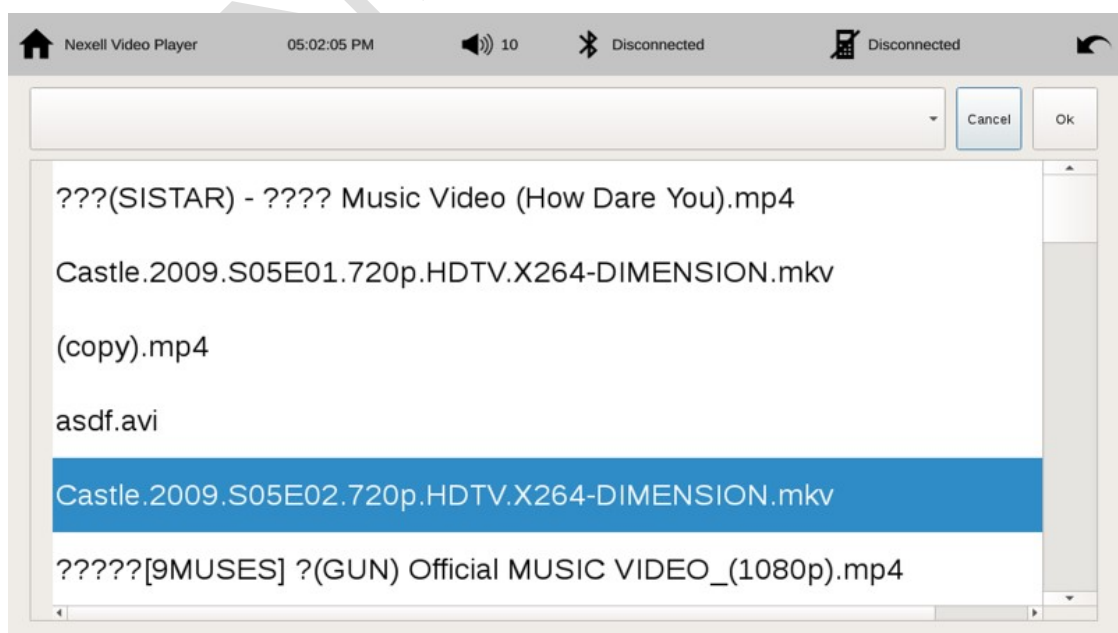
NxVideoPlayer disappears behind other application by changing Z-order.



1.2.4 UI Playlist Details

Playlist UI is shown by clicking playlist button.

List can be scrolled down, and items in list can be played by double-clicking or select(item click) and Ok button.



1.3 Flow

1.3.1 Application Overall

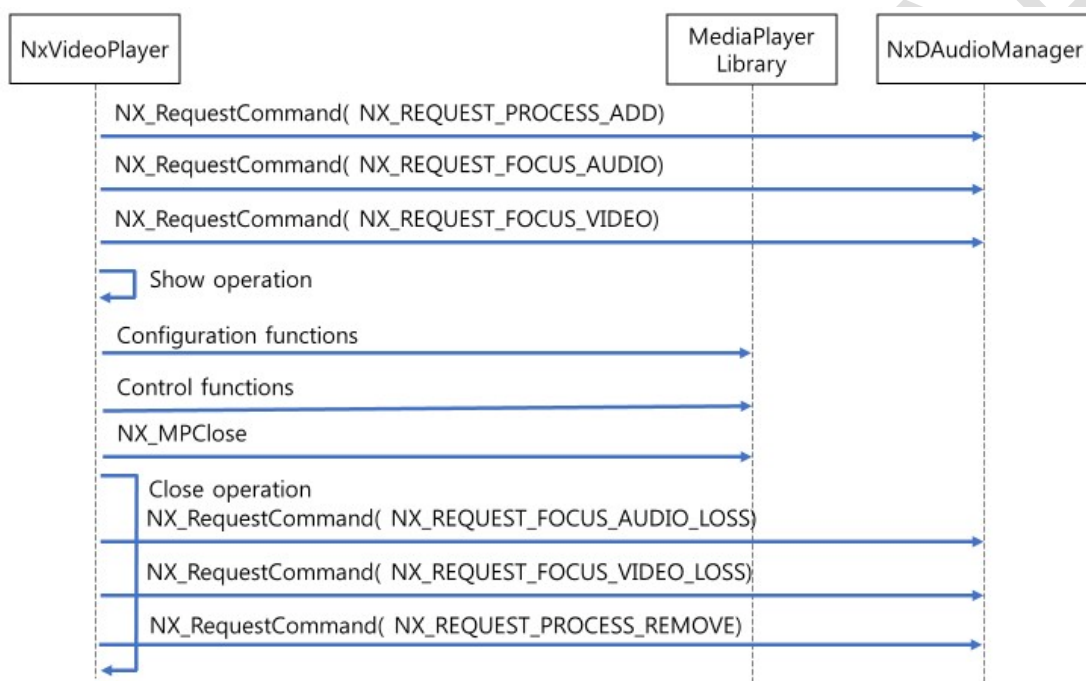
NxVideoPlayer makes IPC server by using libnxdaudioipc to communicate with NxDAudioManager.

All function about IPC in NxVideoPlayer is from libnxdaudioipc.

All NX_RequestCommand function should return NX_REPLY_DONE.

NX_REQUEST_FOCUS_VIDEO is needed, when application wants to appear on screen.

Especially, NX_REQUEST_FOCUS_AUDIO must return NX_REPLY_DONE in order to use Audio Device.

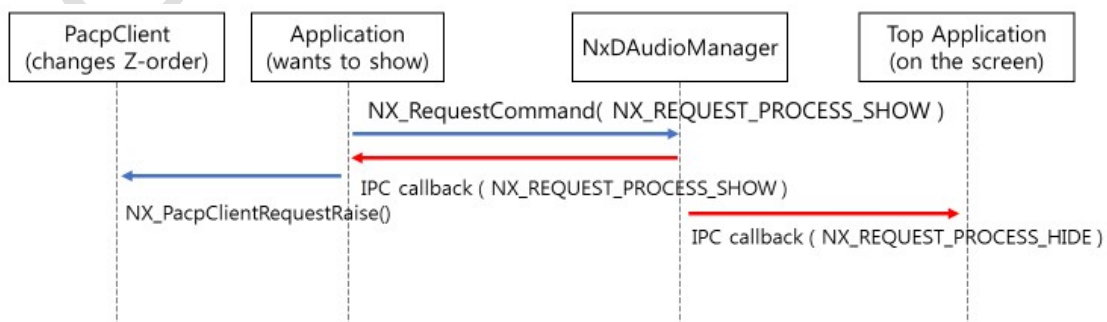


1.3.2 Application Switching

Each application has IPC server to communicate with NxDAudioManager.

PacpClient is used for changing Z-order of application.

IPC callback (NX_REQUEST_PROCESS_SHOW) is pended until NX_PacpClientRequestRaise() is done.



Chap 2. GUI Guide

2.1 Status Bar

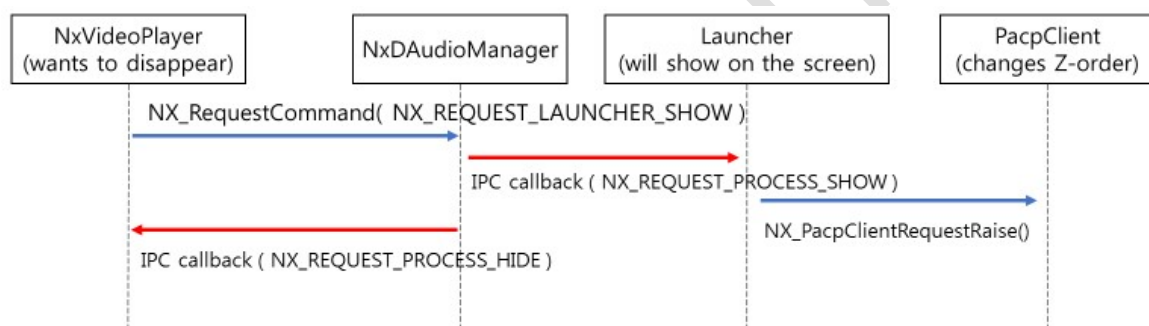
NxStatusBar is used for Status Bar.

2.1.1 Home

When Home button clicked, current application disappears by changing launcher's Z-order to top.

In NxStatusBar's Home button callback function, NxVideoPlayer requests NxDAudioManager to show launcher by using IPC.

2.1.1.1 Flow

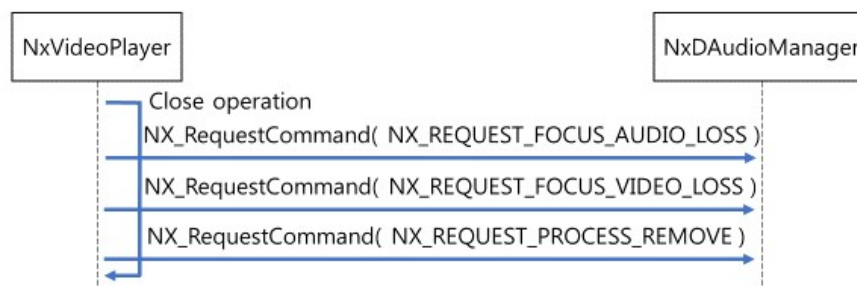


2.1.2 Back

When Back button clicked, NxAudioPlayer closes, and other application(previously top) appears.

In close routine, NxVideoPlayer informs NxDAudioManager of Audio focus loss, Video focus loss, and process remove by using IPC.

2.1.2.1 Flow



2.2 Media Player Control

In NxVideoPlayer application, Media Player library is wrapped by CNX_MoviePlayer class.

2.2.1 Initialization

NxVideoPlayer must get Audio Focus successfully before initialize Media Player lib.

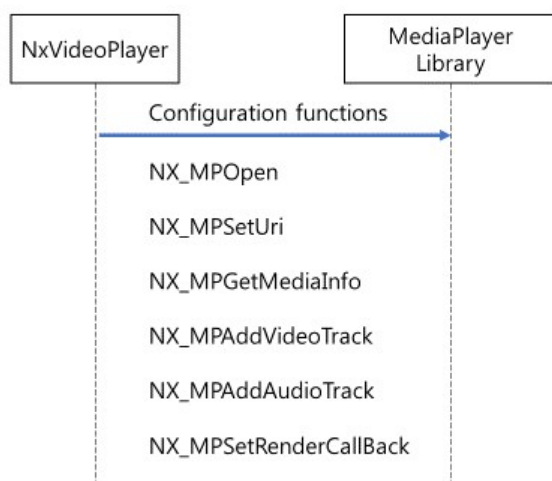
Media Player Initialization is done by CNX_MoviePlayer->InitMediaPlayer

InitMediaPlayer function uses Media Player library's Configuration functions in following order, NX_MPOpen, NX_MPSetUri, NX_MPGetMediaInfo, NX_MPAddVideoTrack, NX_MPAddAudioTrack, and NX_MPSetRenderCallBack.

All Media Player library functions must return MP_ERR_NONE.

NX_MPAddVideoTrack is used for video and NX_MPAddAudioTrack is used for audio.

2.2.1.1 Flow

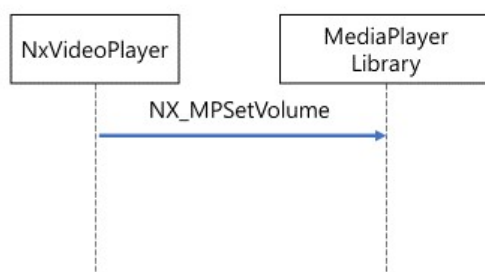


2.2.2 Volume Bar

When Volume Bar is clicked, Bar UI and audio volume are changed to corresponding value.

NX_MPSetVolume in Media Player library is used.

2.2.2.1 Flow

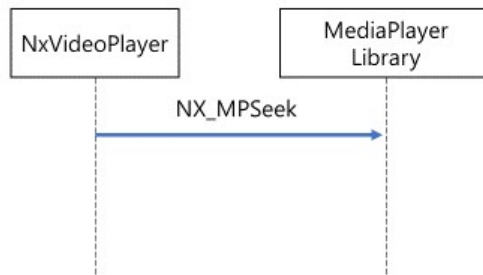


2.2.3 Progress Bar

When Progress Bar is clicked, Bar UI and video position are changed to corresponding value.

NX_MPSeek in Media Player library is used.

2.2.3.1 Flow



2.2.4 Prev Button

When prev button is clicked, NxVideoPlayer stops playing and plays previous video file.

Index of video file list is set to previous video file.

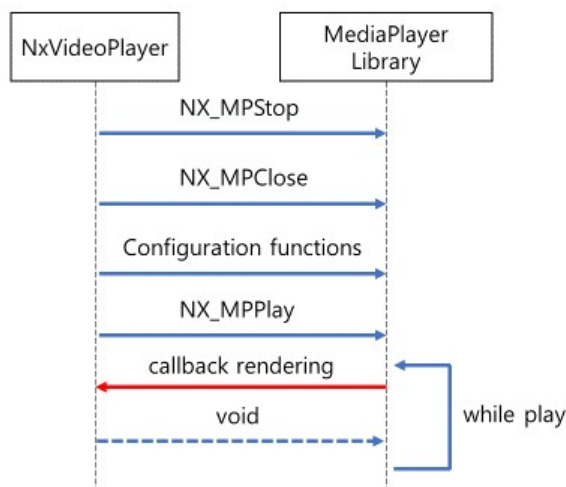
NX_MPStop and NX_MPClose in Media Player library are used.

Initialization is done with previous video file like above explanation.

NX_MPPlay in Media Player library is used after Initialization is succeed.

Media Player library callback rendering will be explained in Play Button.

2.2.4.1 Flow



2.2.5 Play Button

When play button is clicked, NxVideoPlayer plays video file.

In NxVideoPlayer application, play button sequence contains Initialization of Media Player library.

Just after Initialization of Media Player lib is done, NxVideoPlayer updates UI information such as, progress bar, title, and so on.

If Initialization is failed, NX_MPClose in Media Player library is called and Initialization is tried again with next video file in list.

When NxVideoPlayer is already playing video file, play button does nothing.

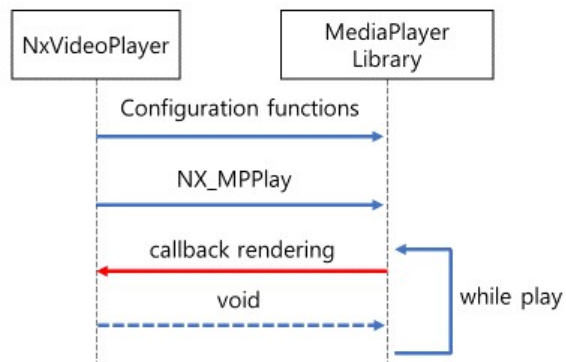
When NxVideoPlayer is in pause, play button just call NX_MPPlay without Initialization.

Media Player Library calls callback function for rendering after NX_MPPlay is called, and NxVideoPlayer draws screen.

Media Player Library calls callback function for rendering is pended until NxVideoPlayer completes drawing screen.

Calling callback function for rendering is repeated while NxVideoPlayer is playing video.

2.2.5.1 Flow

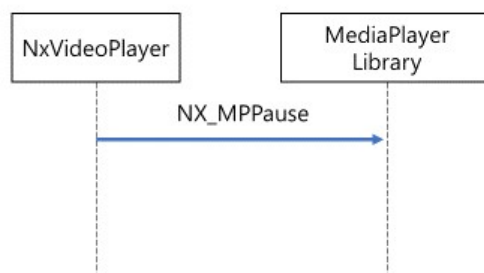


2.2.6 Pause Button

When pause button is clicked, NxVideoPlayer pauses video playing.

NX_MPPause in Media Player library is used.

2.2.6.1 Flow



2.2.7 Next Button

When next button is clicked, NxVideoPlayer stops playing and plays next video file.

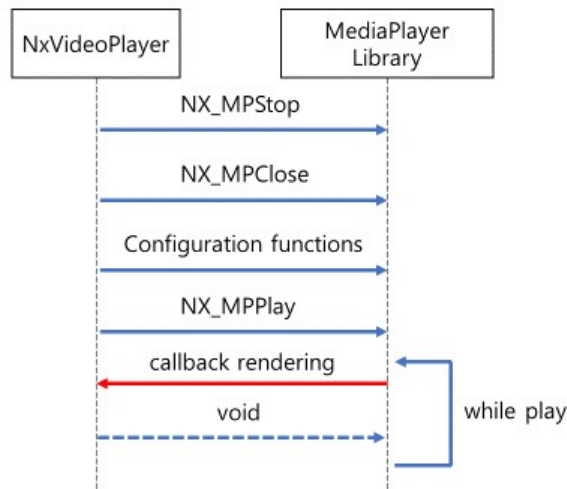
Index of video file list is set to next video file.

NX_MPStop and NX_MPClose in Media Player library are used.

Initialization is done with next video file like above explanation.

NX_MPPlay in Media Player library is used.

2.2.7.1 Flow

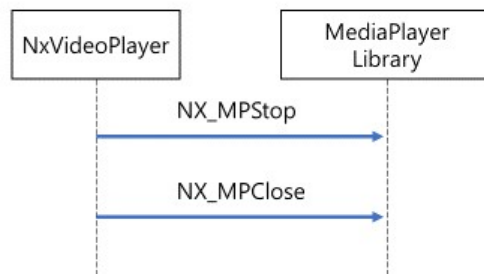


2.2.8 Stop Button

When stop button is clicked, NxVideoPlayer stops playing video.

NX_MPStop and NX_MPClose in Media Player library are used.

2.2.8.1 Flow



2.2.9 Playlist Button

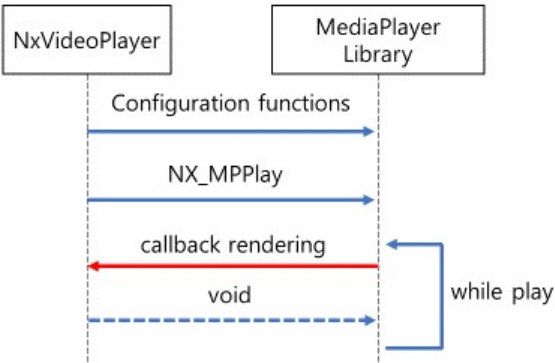
When playlist button is clicked, NxVideoPlayer shows playlist UI.

If NxVideoPlayer was playing video, it continues until some video file is selected.

If file is selected, NxVideoPlayer stops playing video and plays selected video file.

2.2.9.1 Flow

The case that file is selected.



CONFIDENTIAL

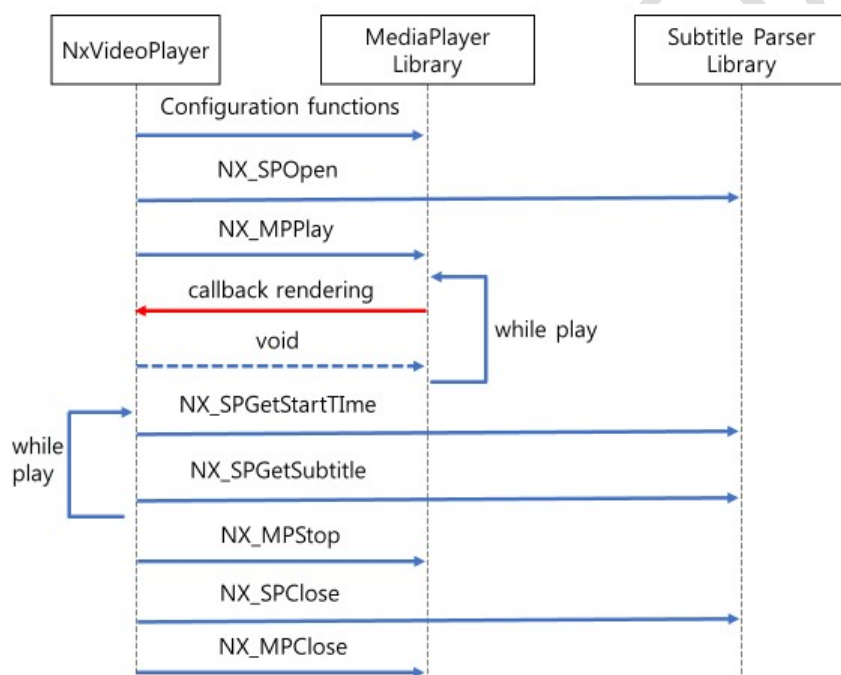
Chap 3. Subtitle

3.1 Overview

In NxVideoPlayer application, Subtitle Parser library is wrapped by CNX_MoviePlayer class.

NX_SPGetStartTIme and NX_SPGetSubtitle are called repeatedly while NxVideoPlayer is playing video.

3.2 Application Overall Flow



Chap 4. Storage Event

4.1 OverView

NxVideoPlayer detects storage events.

Used storage events are removing external storage(USB, SD-card) and media scan done.

Media scan done event is received, whenever storage status is changed such as inserting and removing.

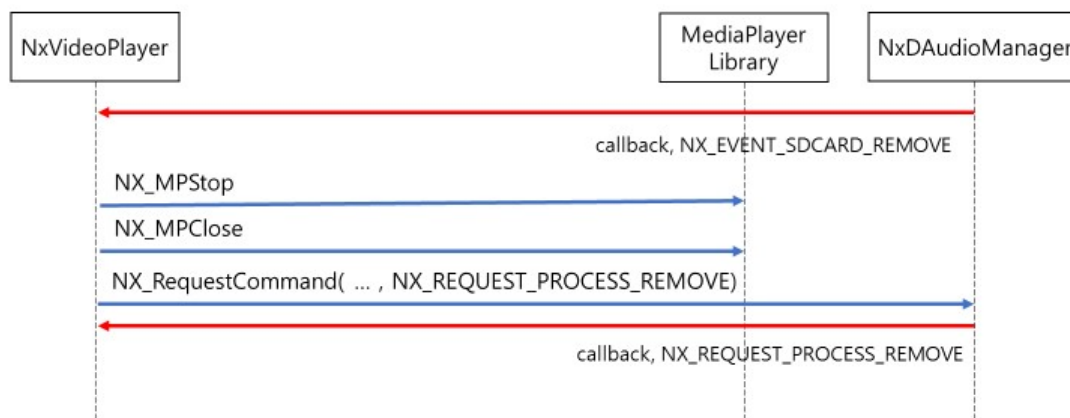
If removing storage event is received, NxVideoPlayer is closed.

If media scan done event is received, NxVideoPlayer refreshes media file list.

4.2 Flow

4.2.1 Removing Storage Event

`NX_EVENT_SDCARD_REMOVE` and `NX_EVENT_USB_REMOVE` are used.



4.2.2 Inserting Storage Event

When `NX_EVENT_MEDIA_SCAN_DONE` is received, media file list is refreshed.



Chap 5. **Known Issues**

5.1 To Do List

Supporting Multi Language for subtitle and playlist UI.

NX_MPStop should be checked again.

Storage Event might be improvable.

CONFIDENTIAL