

SDK API User's Guide (ETC)

Version 0.7.0

Display Audio

Solution Team



Release information

The following changes have been made to this document.

Change History

Date	Change
04 Dec. 2017	First release for v0.6.0
15 Nov. 2018	Second release for v0.7.0

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	Feature.....	1
Chap 2.	Private Configuration	2
2.1	Overview	2
2.2	APIs	2
Chap 3.	SQL	5
3.1	Overview	5
3.2	APIs	5
Chap 4.	NxLog	6
4.1	How to use NX log macro	6
4.2	How to use NX logcat	6
4.3	How to use NX logrotate service	7
Chap 5.	History	8
5.1	Known Issue.....	8

5.2 To Do List.....	8
---------------------	---

CONFIDENTIAL

Chap 1. Overview

1.1 Overview

This document describes SDKs for Display Audio development.

1.2 Feature

The SDK provides as below.

- libnxconfig : private configuration (Android SharedPreference Modeling)
- libnxsqlutils : SQLite database access. (support reading operation)
- libnxdaudioipc : IPC Server / Client Engine, IPC engine wrapper

Chap 2. Private Configuration

2.1 Overview

The libnxconfig provides to manage application private configuration. This library is not thread-safe. So it is not suitable to share configuration with application.

2.2 APIs

2.2.1 Open()

```
virtual int32_t Open(
    const char *pFile
) = 0;
```

Description

Open XML configuration file.

Parameter

-. pFile : XML Configuration file.

Parameter

0 is successful. -1 is failed.

2.2.2 Close()

```
virtual void Open(
    void
) = 0;
```

Description

Close XML configuration file.

Parameter

None.

Parameter

None.

2.2.3 Write()

```
virtual int32_t Write(
    const char *pKey,
    char *pValue
```

) = 0;
Description Write Configuration.
Parameter -. pKey : configuration key. -. pValue : configuration value.
Parameter 0 is successful. -1 is failed.

2.2.4 Read()

virtual int32_t Read(const char *pKey, char **ppValue) = 0;
Description Read Configuration.
Parameter -. pKey : configuration key. -. ppValue : configuration value.
Parameter 0 is successful. -1 is failed.

2.2.5 Remove()

virtual int32_t Read(const char *pKey) = 0;
Description Remove Configuration.
Parameter -. pKey : configuration key.
Parameter 0 is successful. -1 is failed.

2.2.6 Dump()

virtual void Dump(void) = 0;
Description Dump configuration for debugging.
Parameter

None.
Parameter
None.

CONFIDENTIAL

Chap 3. SQL

3.1 Overview

The SQLite wrapper library provides to read database. This library designs to query table in database. The SQLite is database based file. So this library ensures data integrity.

3.2 APIs

3.2.1 NX_SQLiteGetData()

```
int32_t NX_SQLiteGetData(
    const char *pDatabase,
    const char *pTable,
    int32_t (*cbFunc)(void*, int32_t, char**, char**),
    void *pObj = NULL,
) = 0;
```

Description

Access database using SQLite.

Parameter

-. pDatabase : database name.

-. pTable : table name.

-. cbFunc : result data callback

int32_t cbFunc(void *pObj, int32_t iColumnNum, char **ppColumnValue, char **ppColumnName)

-. pObj : private handle

-. iColumnNum : column number of table.

-. ppColumnValue: column value of table.

-. ppColumnName: column name of table.

-. pObj : private handle.

Return Value

0 is successful, -1 is failed.

Chap 4. NxLog

4.1 How to use NX log macro

```
#define LOG_TAG "Your tag name"
#include <NX_Log.h>

NXLOGV("Verbose : This function is %s", __func__);
NXLOGD("Debug : This function is %s", __func__);
NXLOGN("Notification : This function is %s", __func__);
NXLOGI("Information : This function is %s", __func__);
NXLOGW("Warning : This function is %s", __func__);
NXLOGE("Error : This function is %s", __func__);
```

4.2 How to use NX logcat

```
$ adb logcat -h
```

Usage:

On target => nxlogcat or logcat <option> <value>

ex> nxlogcat -t -p -l 5

On pc => adb logcat or adb shell logcat or adb shell nxlogcat <option> <value>

ex> adb logcat -t -p -l 5

Common options:

-h: help

-v: print versions (macro header, this program)

Print options:

-c: clear the log

-t: print with time

-p: print with PID

-f: print only the current log

-x: print without system log (kernel, systemd, etc...)

-l <level>: print the log up to the specified level

Level:

0 : error

1 : warning

```

2 : notice
3 : information
4 : debug
5 : verbose

```

4.3 How to use NX logrotate service

```
$ nxlogrotate -h
```

Usage:

```
nxlogrotate <option> <value>
```

```
ex> nxlogrotate -s 100k -r 5 -p /tmp
```

```
ex> nxlogrotate -s 1m -r 5 -p /tmp
```

Common options:

```
-h: help
```

```
-v: print versions
```

Settings options:

```
-s: log limitation size (10k ~ 10m), default 1m
```

```
-r: log rotation count (0 ~ 10), default 5
```

```
-p: backup log path
```

4.3.1 Run service (Default)

```
$ vi /lib/systemd/system/nxlogrotate.service
```

```
[Unit]
```

```
Description=NX log rotate daemon
```

```
Wants=syslog.service
```

```
After=syslog.service
```

```
[Service]
```

```
Type=idle
```

```
ExecStart=/usr/sbin/nxlogrotate
```

```
[Install]
```

```
WantedBy=multi-user.target
```

```
$ ln -s /lib/systemd/system/nxlogrotate.service /lib/systemd/system/multiuser.target.wants/nxlogrotate.service
```

Chap 5. **History**

5.1 Known Issue.

-. Not Yet.

5.2 To Do List

-. Not Yet.

CONFIDENTIAL