

www.dokodemoterebi.com

Android Player Software

API Reference

Issue: 1.0.0

Date: 2012.11.22

About This Document

Purpose

This document describes the related product versions, intended audience and document contents.

Related Versions

The following table lists the product version related to this document.

Product Name	Version
Further Android Software Development Kit (SDK)	V1.0.0

Intended Audience

This document describes the reference information developed based Further Android, and it is intended for the programmers who meet the following requirements:

- Be familiar with JAVA programming language
- Be familiar with basic Android SDK invoking

Organization

The document first summarizes the Further Android API function types and their connections, and then describes the using method of each function in detail.

This document is organized as follows:

Chapter	Content
1 Overview	Describes the components and development environments of Further Android SDK.
2 API Description	Provides an overview of Further Android SDK API, and describes all the API functions in detail.
3 Data Structures Description	Describes the enumerated types used by Further Android.
4 API Application Instances (userlibs2)	Describes Further Android application instance such as connecting Box, getting AV data and using RTSP or HTTP Live Streaming protocol in local player to play AV streams.

1 Overview

Further Android network library provides the file FurtherNet-mobile.jar. The library is mainly used for interactive communication with 8960Box, including receiving audio/video data and sending data through RTSP server to play AV streaming and send commands to control Box.

1.1 FurtherNet-mobile.jar importing method (in Eclipse environment)

- 1.1.1 Create "lib" folder in the Android project root directory, and then copy the file FurtherNet-mobile.jar to the folder.
- 1.1.2 Start Eclipse, right click the Android project, select "Build Path->Configure Build Path...".
- 1.1.3 In the "Java Build Path" dialog box, select "Java Build Path->Libraries->Add JARs...".
- 1.1.4 In the "JAR Selection" dialog box, select the path which you created just now "lib->FurtherNet-mobile.jar", Click "OK".
- 1.1.5 If it appears "Referenced Libraries" in the project, it means you create successfully. Then you only need to import com.Further.MobileNet.* and instantiate the interface class Net in the library to use the library with Net object.
- 1.1.6 If using the HTTP Live streaming protocol in local player to play AV streaming, please copy the file libhttplivesream.so to the root directory /libs/armeabi/.

1.2 Function list

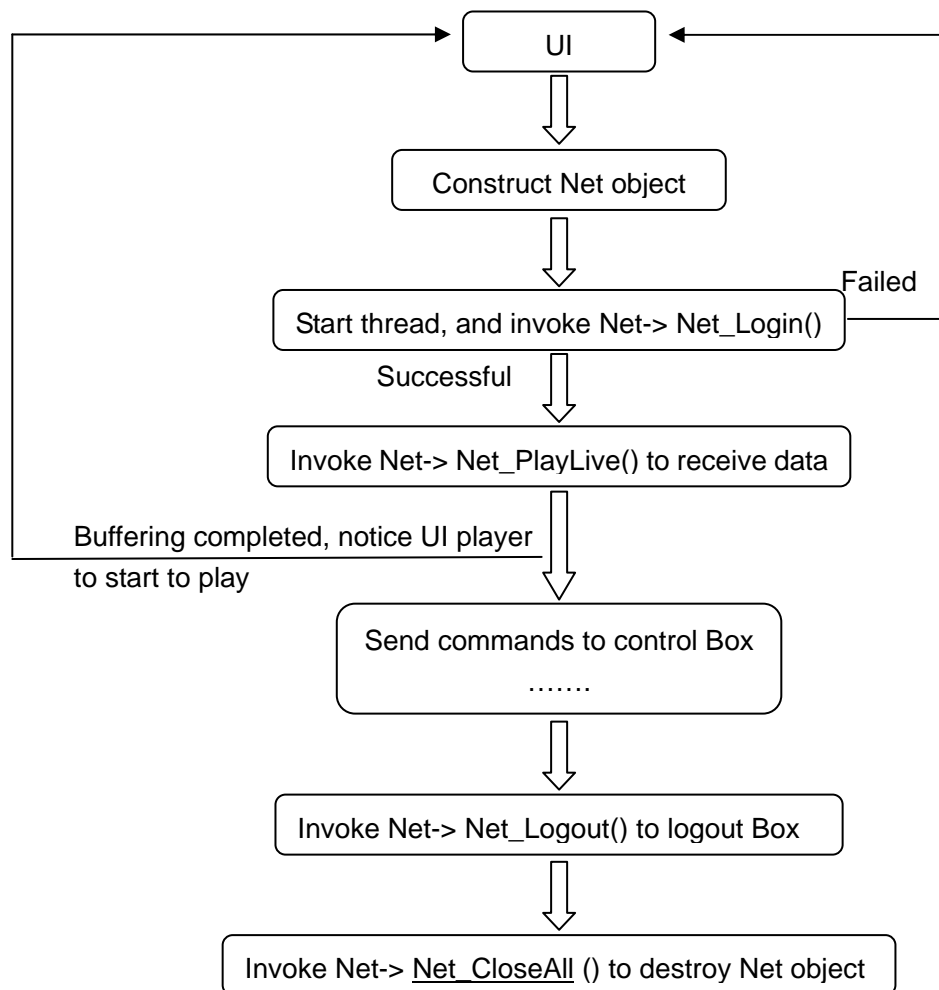
1.2.1 UI Interface functions provided by Net class

Function name	Description
Net	Constructor
Net_CloseAll	Destroys the Net object.
Net_Login	Login Box
Net_Logout	Logout Box
GetBitRate	Gets the current bitrate.
GetBufferSize	Gets the current buffer size.
Net_GetNetStat	Gets the network state.
Net_PlayLive	Starts to play data.
SetResolution	Sets resolution.
SetSystemCategory	Sets TV system.
initRtspServer	Initializes RTSP server.
clearRB	Clears buffers.
Net_ReCreateHttp	Restart the HTTP server.
registerGetData_callback	Registers and gets the AV data objects.

1.2.2 Controlling functions provided by Net class

Function name	Description
Net_CmdIRNormalPress	IR command
Net_CmdIRNormalPress_A7	IR command only used by A7
Net_CmdSetSource	Command to switch video source.
Net_CmdSetVideoFormat	Command to set the resolution.
Net_CmdTunerGetFRQTable	Command to get the channel table.
Net_CmdTunerSetFRQ	Command to switch channel.
Net_CmdGetScheduleInfo	Command to get the schedule recording tasks.
Net_CmdSendAddRecordTask	Command to add or edit a schedule recording task.
Net_CmdSendDelTask	Command to delete the schedule recording task.

1.3 UI invoking flow



2 API Function Description

2.1 FurtherNet-mobile library functions

2.1.1 Net (Context context, Handler handler,String SvrIP1,String SvrIP2,String CMSIP1,String CMSIP2,String TopCMSIP1,String TopCMSIP2,FG_UI_LogFile logFile);

Purpose:	Constructor of network module, used to create the network object.
Parameter:	Input: context: Context Handler: Handle for UI to process messages. SvrIP1: The main server used by 8960. SvrIP2: The backup server used by 8960. CMSIP1: The main CMS server used by A series. CMSIP2: The backup CMS server used by A series. TopCMSIP1: The main top CMS server used by A series. TopCMSIP2: The backup top CMS server used by A series. logFile: Log file object
Return Value:	NULL
Note:	NULL

2.1.2 int Net_CloseAll();

Purpose:	Destroys the network module object.
Parameter:	NULL
Return Value:	NULL
Note:	NULL

2.1.3 LOGINSTATE Net_Login(String in_BoxSN,String in_Password,Net_CommonDefine LOGIN_PARAM);

Purpose:	Login Box procedure function
Parameter:	Input: in_BoxSN: Box SN in_Password: Box login password in_LoginParam: Information structure used when login.
Return Value:	LOGINSTATE (Please refer to Chapter 3 for its description.)
Note:	The function is a block process.

2.1.4 int Net_Logout();

Purpose:	Sends logout command.
Parameter:	NULL
Return Value:	0: Successful; Not 0: Failed
Note:	NULL

2.1.5 int Net_CmdIRNormalPress(int in_Code);

Purpose:	Sends IR learning commands.
Parameter:	IR key value
Return Value:	0: Successful; Not 0: Failed
Note:	NULL

2.1.6 int Net_CmdSendAddRecordTask(int nAddOrUpdateFlag,byte[] recordTask);

Purpose:	Sends command to add or edit a schedule recording task.
Parameter:	nAddOrUpdateFlag: 1: add 2: update recordTask: Schedule recording task data
Return Value:	0: Successful; Not 0: Failed
Note:	NULL

2.1.7 int Net_CmdSendDelTask();

Purpose:	Sends command to delete the schedule recording task.
Parameter:	nTaskID: Task ID
Return Value:	0: Successful; Not 0: Failed
Note:	NULL

2.1.8 int Net_CmdIRNormalPress_A7(int in_Code,int nSourceID);

Purpose:	Sends IR learning command for A7.
Parameter:	In_Code: IR key value nSourceID: 3: Used by CAM source; 0: Used by AV source
Return Value:	0: Successful; Not 0: Failed
Note:	NULL

2.1.9 int Net_CmdSetSource(int in_Source,int in_CancelRecord);

Purpose:	Sends video source switch command.
Parameter:	In_Source: Video source ID 0: AV; 2: TV; 3: CAM In_CancelRecord: Recording cancel mark (The destination source is recording when switching source) 0: Don't cancel 1: Cancel
Return Value:	0: Successful; Not 0: Failed
Note:	NULL

2.1.10 int Net_CmdSetVideoFormat(int in_Video_Size);

Purpose:	Sends resolution switch command.
Parameter:	In_Video_Size: Resolution ID 0: D1; 1: HD1; 2: CIF; 3: ECO In_CancelRecord: Recording cancel mark (The destination source is recording when switching source) 0: Don't cancel 1: Cancel
Return Value:	0: Successful; Not 0: Failed
Note:	NULL

2.1.11 int Net_CmdTunerGetFRQTable();

Purpose:	Sends command to get channel table.
Parameter:	NULL
Return Value:	0: Successful; Not 0: Failed
Note:	NULL

2.1.12 int Net_CmdTunerSetFRQ(int in_ChNum,int in_FreqNum);

Purpose:	Sends channel switch command.
Parameter:	In_ChNum: Channel number In_FreqNum: Frequency value
Return Value:	0: Successful; Not 0: Failed
Note:	NULL

2.1.13 int Net_CmdGetScheduleInfo();

Purpose:	Sends command to get schedule recording tasks.
Parameter:	NULL
Return Value:	0: Successful; Not 0: Failed
Note:	NULL

2.1.14 int GetBitRate ();

Purpose:	Gets the current bitrate.
Parameter:	NULL
Return Value:	Bitrate value
Note:	NULL

2.1.15 int GetBufferSize();

Purpose:	Gets the current buffer size.
Parameter:	NULL
Return Value:	Size value
Note:	NULL

2.1.16 void initRtspServer();

Purpose:	Initializes various state values of RTSP server.
Parameter:	NULL
Return Value:	NULL
Note:	NULL

2.1.17 void clearRB(int nClearFlag);

Purpose:	Clears buffers.
Parameter:	nClearFlag: 0: Clears buffers when switching channel 1: Clears buffers when switching source. 2: A8 clears buffers when switching channel.
Return Value:	NULL
Note:	NULL

2.1.18 void SetSystemCategory(int nSC);

Purpose:	Sets TV system.
Parameter:	nSC: 0: NTSC 1: PAL
Return Value:	NULL
Note:	NULL

2.1.19 void SetResolution (int bFlag);

Purpose:	Sets resolution.
Parameter:	bFlag: 0: D1; 1: HD1; 2: CIF; 3: ECO
Return Value:	NULL
Note:	NULL

2.1.20 void Net_SetNetStat(NETSTATE in_Stat);

Purpose:	Sets the network module state.
Parameter:	In_Stat: (Please refer to Chapter 3 for its description.)
Return Value:	NULL
Note:	NULL

2.1.21 NETSTATE Net_GetNetStat();

Purpose:	Gets the network module state.
Parameter:	NULL
Return Value:	(Please refer to Chapter 3 for its description.)
Note:	NULL

2.1.22 int Net_PlayLive (int nProtocol);

Purpose:	Starts to receive data and play.
Parameter:	0: Only returns AV data 1: Using RTSP protocol in local player to play. 2: Using HTTP Live Streaming protocol in local player to play.
Return Value:	0: Successful; Not 0: Failed
Note:	NULL

2.1.23 void registerGetData_callback (GetAVData_Callback cb);

Purpose:	Registers and gets the AV data objects to Net module.
Parameter:	Rewrite GetAVData_Callback interface object
Return Value:	NULL
Note:	NULL

2.1.24 void Net_ReCreateHttp ()

Purpose:	Restart the HTTP server.
Parameter:	NULL
Return Value:	NULL
Note:	when using HTTP Live Streaming protocol to play, HTTP server must be restarted after the successful resolution switch.

3 Data Structures Description

3.1 Return value of Net_Login command

```
enum LOGINSTATE
{
    LOGINSTATE_INIT,
    LOGINSTATE_OK,
    LOGINSTATE_LINKSVRFAILED,
    LOGINSTATE_NORTSPACK,
    LOGINSTATE_BOXSNERROR,
    LOGINSTATE_BOXNOTINLINE,
    LOGINSTATE_CHANGECMSSVR,
    LOGINSTATE_HPLAYFAILED,
    LOGINSTATE_SENDPLAYFAILED,
    LOGINSTATE_HDESCRIBEFAILED,
    LOGINSTATE_SENDDDESCRIBEFAILED,
    LOGINSTATE_SVRCONNECTFAILED,
    LOGINSTATE_PARSEURLFAILED,
    LOGINSTATE_CONNECT2PORTFAILED,
    LOGINSTATE_SENDCONNECTBOXFAILED,
    LOGINSTATE_PAWERROR,
    LOGINSTATE_SNUNREGIST,
    LOGINSTATE_USERTYPEPEERROR,
    LOGINSTATE_NULLSVR,
    LOGINSTATE_OFFUSE,
    LOGINSTATE_NOFREE,
    LOGINSTATE_NOINTERNET,
    LOGINSTATE_DIRECTERROR,
    LOGINSTATE_CONNECTBOXFAILED,
    LOGINSTATE_UNKNOWERR,
    //=====MINI SYSTEM USE=====
    LOGINSTATE_MINSYSTEMUSERID,    // ID not registered
    LOGINSTATE_MINSYSTEMDISABLE,   // ID is disabled.
    LOGINSTATE_MINSYSTEMMATURITY,  // ID has expired
    LOGINSTATE_MINSYSTEMPASSERR,   // The password is wrong
    LOGINSTATE_MINSYSTEMSVRBUSY,   // SBox is busy
    LOGINSTATE_MINSYSTEMNOFREE,    // There's no free Box.
    LOGINSTATE_MINSYSTEMREPEAT;    // Repeated account login
}
```

3.2 Network state value

enum NETSTATE

```
{  
    NETSTATE_INIT ,  
    NETSTATE_LOGINBOXOK ,  
    NETSTATE_LOGINBOXFAILED ,  
    NETSTATE_PLAYING ,  
    NETSTATE_UNKNOWERR ,  
    NETSTATE_PLAYFAILED ;  
}
```

3.3 Parameter type used when login

public static class LOGIN_PARAM

```
{  
    public int nLastSource;    // Last logout video source  
    public int nLastChnID;    // Last logout channel ID number  
    public int nLastFreq;    // Last logout frequency value  
    public int nLinkType;    // 0: Connect Box as normal; 1: Using last successful IP to  
                             // connect; 2: Using custom IP to connect  
    public int nLoginFlag;    // Login for the first time: 0; Login for the second time: 1  
    public int nLanSearch;    // Dynamic password used by T2 (Needn't enter it when login)  
    public int nLoginApp;    // 0: AP; 1: Schedule  
    public String strDefP2PS1; // The default IP of main P2P server, used by 8960,A2,A5,A7,A8  
    public String strDefP2PS2; // The default IP of backup P2P server  
  
    public String strDefSBoxIP1; // The default IP of main T1 server, used by MINI SYSTEM.  
    public String strDefSBoxIP2; // The default IP of backup T1 server  
  
    public String strLastIP;    // Box IP of the last successful login, used for direct connection.  
    public int nLastPort;    // Box Port of the last successful login  
    public String strCustomIP; // Custom Box IP  
    public int nCustomPort;    // Custom Box Port  
}
```

4 API Application Instances

Please refer to appendix for UseNetLib2 source code.

www.dokodemoterebi.com