

OpenH264 build and update for wme

Table of Contents

OpenH264 build and update for wme.....	1
1. Build and update	2
1.1 Basic flowchart	2
1.2 Script for build and update.....	3
1.3 Check previous build/update history.....	3
1.4 Build and update.....	4
1.4.1 basic flow chart.....	4
1.4.2 script	6
1.4.3 Example:	7
2. Download from ftp server	9
2.1 Basic info	9
2.1 Basic flowchart	10
2.2 Script	11
2.2.1 check version	11
2.2.1 download	12
3. To do	13

1. Build and update

1.1 Basic flowchart

Unify option name for skip build and download from ftp server is:

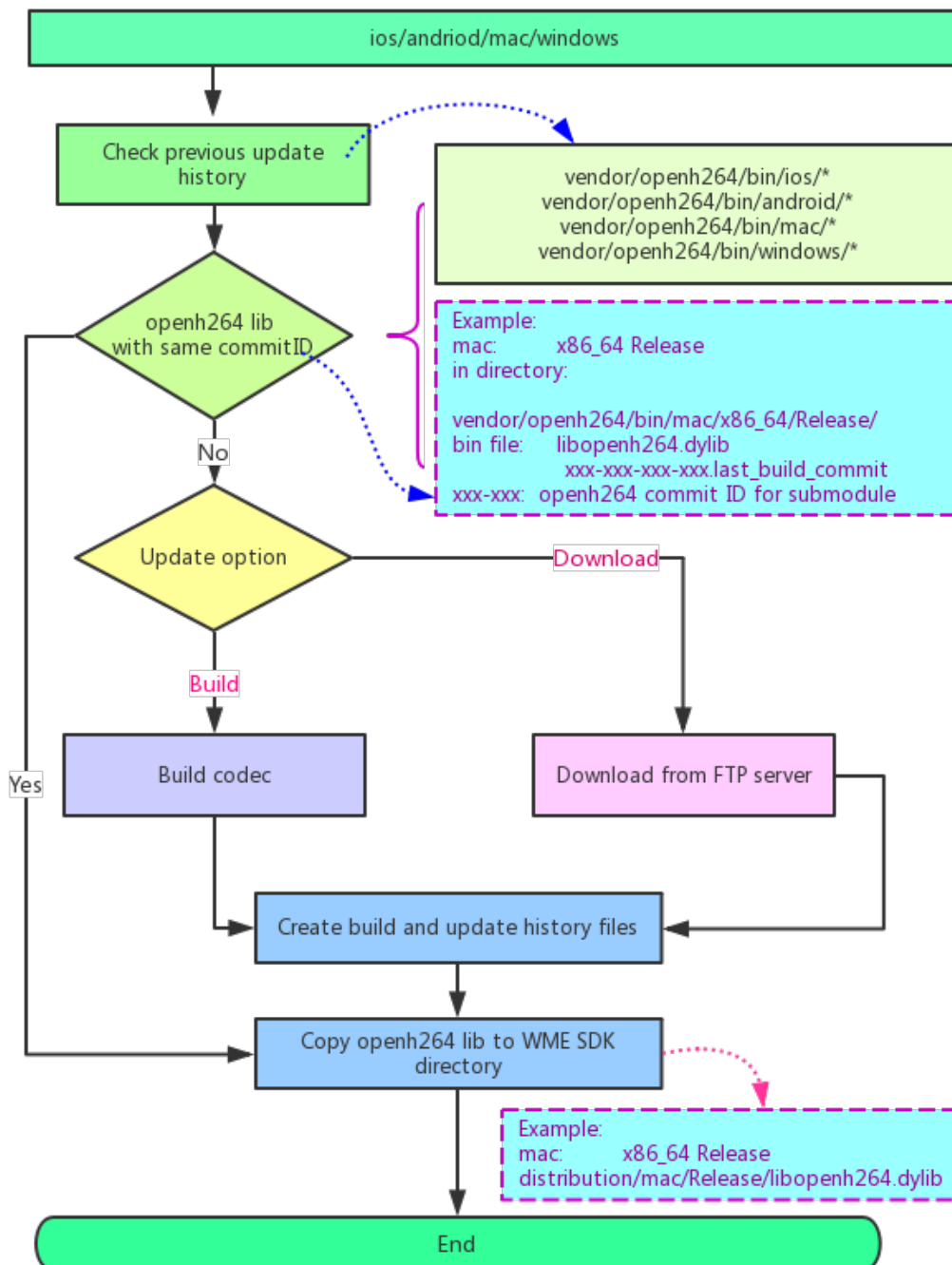
openh264-skip-build

Example:

for mac build, go to wme/build/mac and run:

sh build.sh openh264-skip-build

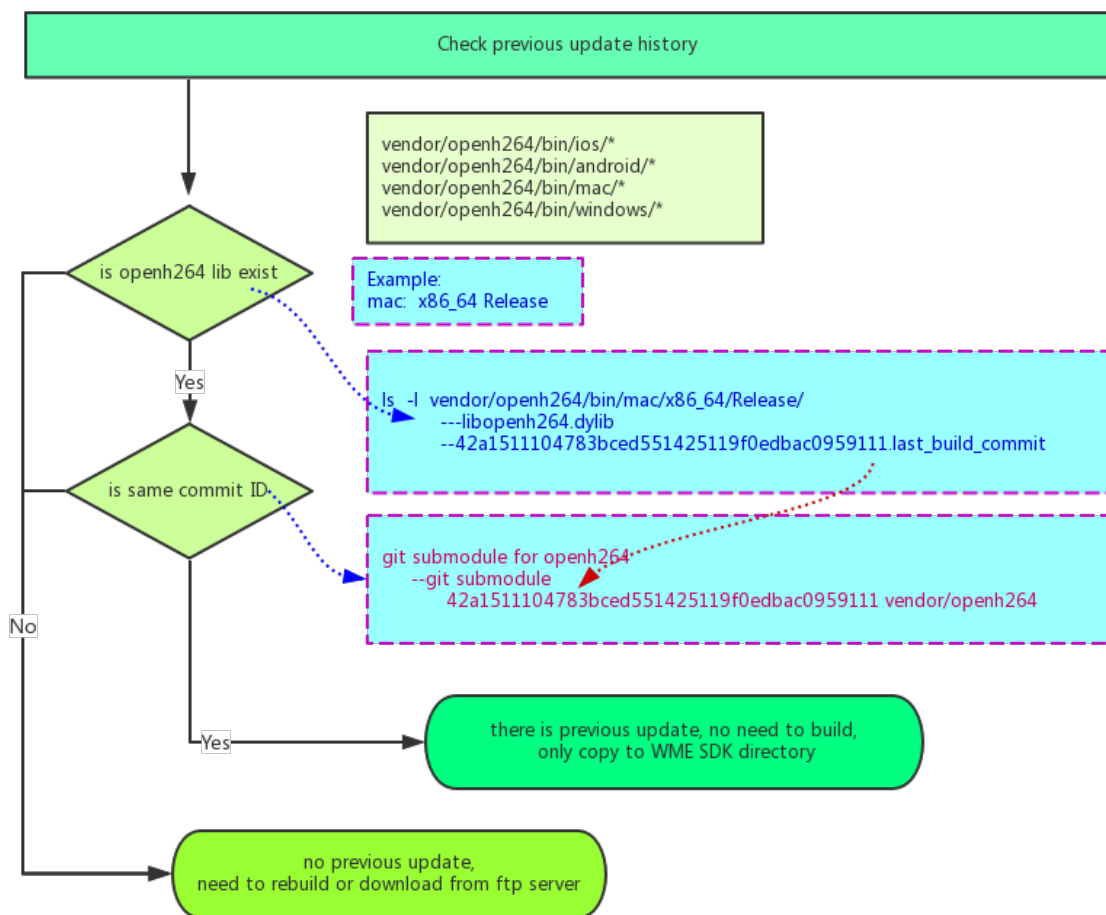
script will skip openh264 build and download from ftp server.



1.2 Script for build and update

- ✚ build/ios/buildAndupdate_openh264.sh
- ✚ build/mac/buildAndupdate_openh264.sh
- ✚ build/windows/build_openh264.py
- ✚ mediaengine/shark/bld/client/android/build.py
- ✚ build/linux/build.sh

1.3 Check previous build/update history



Example: for mac, in script file: `build/mac/buildAndupdate_openh264.sh`

Check openh264 commit ID

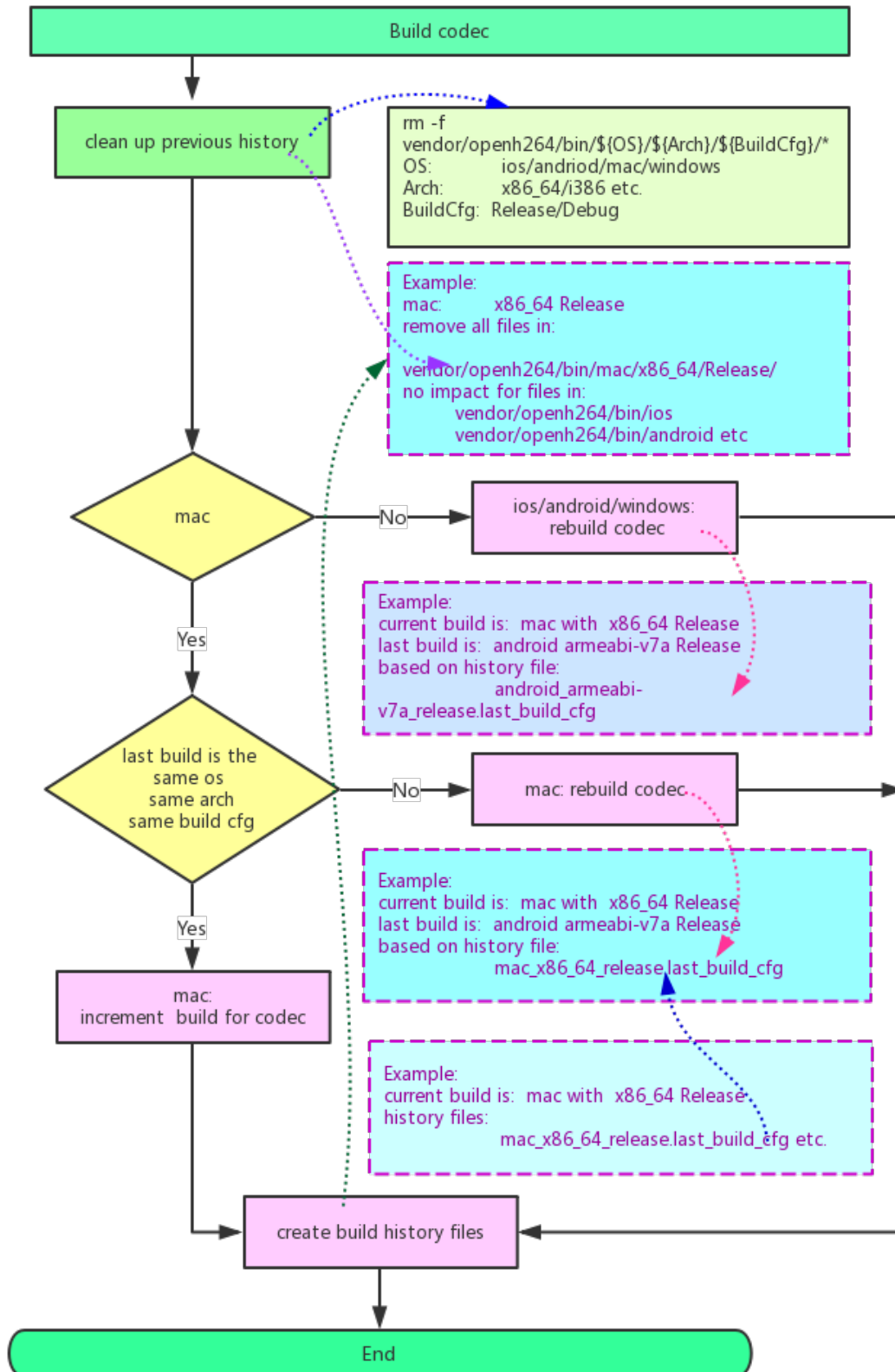
```
LibVersion=`git submodule | grep "openh264" | awk '{print $1}'`
LibVersionInfo="${BinDir}/${LibVersion}.last_build_commit"
```

Check openh264 lib exist or not, with the same commit ID

```
cd ${CODEC_PROJECT_PATH}
[ -e "${BinFile}" ] && [ -e "${LibVersionInfo}" ] && LastBuildFlag="True"
#clean bin dir if no last build or last build version not match current commit ID
[ "${LastBuildFlag}" = "False" ] && [ -d "${BinDir}" ] && rm -rf "${BinDir}"
```

1.4 Build and update

1.4.1 basic flow chart



1.4.2 script

Example: for mac, in script file: [build/mac/buildAndupdate_openh264.sh](#)

For mac,

- ✚ if last build is also for mac, not for ios/android etc.
- ✚ last build is the same arch : x86_64 or i386 etc
- ✚ last build is the same build cfg: Release or Debug

increment build only, for mac case, only using command:

- ✚ make
- ✚ no make clean

```
[ -e "${LastBuildCfg}" ] && BuildCfgFlag="True"
cd -
```

```
#for case that there is previous build libs
if [ "${LastBuildFlag}" = "True" ]; then
[ "$BuildDownload" = "build" ] && [ "${BuildCfgFlag}" = "True" ] && BuildOpenh264
else
[ "$BuildDownload" = "build" ] && CleanOpenH264 && BuildOpenh264
[ "${BuildDownload}" = "skip-build" ] && DownloadOpenH264
fi
```

```
make ARCH=${Arch} BUILDTYPE=${Config} >Mac_Build.log
if [ $? -ne 0 ]; then
cat Mac_Build.log
echo "build error, please check"
exit 1
fi
```

1.4.3 Example:

git submodule command to get openh264 commit ID

```
[HUASHI-M-400W:ios huashi$ git submodule
fd2c8e4b7a54cc6c722fe2f0c6b5f0e748868e56 ../../vendor/AudioPairing
d959d7095468d42b8fb5d6f4f111558dbcb936 ../../vendor/code-style (
555bd944e509174947e5b8fd5860436dbb6795aa ../../vendor/libsdp (remo
12ea269db3e4e9af7ce5de524fe07312fa6e035b ../../vendor/libsrtp (v2.
84e88c1bb983934d30980f97c3214475a22c2acd ../../vendor/mari (v1.0.4
42a1511104783bcd551425119f0edbac0959111 ../../vendor/openh264 (he
```

Build for ios with dev/release

```
arch info for final openh264 lib:
Architectures in the fat file: Release-iphoneos/libopenh264.a are: armv7 armv7s arm64
/Users/huashi/project/WME/Huade/build/ios
*****
copy openh264 lib to wme distribution dir
cp -f ../../vendor/openh264/Release-iphoneos/libopenh264.a ../../distribution/ios/Release-iphoneos/
*****
copy debug lib for wme-debug build: cp -f ../../vendor/openh264/Release-iphoneos/libopenh264.a ../../
*****
Add build/update history for next build check
BinFile is: bin/ios/dev/Release/libopenh264.a
LibVersionInfo is: bin/ios/dev/Release/42a1511104783bcd551425119f0edbac0959111.last_build_commit
*****
/Users/huashi/project/WME/Huade/build/ios

No previous build history, rebuild libs for wme successfully
*****build/update lib for wme iOS successfully*****
```

```
ff 150 May 4 15:32 Release
ls -l ../../vendor/openh264/bin/ios/dev/Release/

ff 0 May 4 15:32 42a1511104783bcd551425119f0edbac0959111.last_build_commit
ff 9668800 May 4 15:32 libopenh264.a
ls -l ../../vendor/openh264/

aff 257 Feb 28 13:28 CODING_STYLE
aff 598 Feb 28 13:28 CONTRIBUTORS
aff 1295 Feb 28 13:28 LICENSE
aff 13120 Apr 24 14:26 Makefile
aff 8101 Feb 28 13:28 README.md
aff 8875 Feb 28 13:28 RELEASES
aff 204 May 4 15:32 Release-iphoneos
aff 170 Feb 28 13:29 autotest
aff 102 May 4 15:32 bin
aff 748 Apr 24 14:26 build
aff 167 Feb 28 13:29 code-coverage.sh
aff 306 Feb 28 13:29 codec
aff 170 Feb 28 13:29 docs
aff 116 Feb 28 13:29 gmpopenh264.info
aff 82049 May 4 15:31 iOS_Build.log
aff 19790 May 4 15:30 iOS_clean.log
aff 0 May 4 15:32 ios_dev_Release.last_build_cfg
aff 270648 May 4 15:31 libcommon.a
```

Build for mac with x86_64/Release

```
/Users/huashi/project/WME/Huade/build/mac
*****
copy openh264 lib to distribution dir
cp -rf ../../vendor/openh264/libopenh264.dylib ../../distribution/mac/Release
*****
copy debug lib for wme-debug build: cp -f ../../vendor/openh264/libopenh264.dylib ../../distribution/mac/Debug
*****
Add build/update history for next build check
BinFile is: bin/mac/x86_64/Release/libopenh264.dylib
LibVersionInfo is: bin/mac/x86_64/Release/42a1511104783bcd551425119f0edbac0959111.last_build_commit
*****
/Users/huashi/project/WME/Huade/build/mac











No previous build history, rebuild libs for wme successfully
*****build/update lib for wme mac successfully*****
HUASHI-M-400W:mac huashi$ ls -l ../../vendor/openh264/bin/mac/
total 0
drwxr-xr-x  3 huashi  staff   102 May  4 15:40 x86_64
HUASHI-M-400W:mac huashi$ ls -l ../../vendor/openh264/bin/mac/x86_64/Release/
total 2256
-rw-r--r--  1 huashi  staff      0 May  4 15:40 42a1511104783bcd551425119f0edbac0959111.last_build_commit
-rwxr-xr-x  1 huashi  staff 1152200 May  4 15:40 libopenh264.dylib
HUASHI-M-400W:mac huashi$ ls -l ../../vendor/openh264/
total 11472
-rw-r--r--  1 huashi  staff    257 Feb 28 13:28 CODING_STYLE
-rw-r--r--  1 huashi  staff    598 Feb 28 13:28 CONTRIBUTORS
-rw-r--r--  1 huashi  staff   1295 Feb 28 13:28 LICENSE
-rw-r--r--  1 huashi  staff  47306 May  4 15:40 Mac_Build.log
-rw-r--r--  1 huashi  staff  20388 May  4 15:39 Mac_Clean.log
-rw-r--r--  1 huashi  staff   13120 Apr 24 14:26 Makefile
-rw-r--r--  1 huashi  staff      0 May  4 15:40 OSX_x86_64_Release.last_build_cfg
```








2. Download from ftp server















2.1 Basic info

- FTP server: <ftp://10.224.203.70/JenkinsRelease/>
Account: [OpenH264/wme@cisco](#)
- FTP server wiki: <https://wiki.cisco.com/display/WX2/OpenH264+ftp+server>
- Tools for upload and download from ftp server
<https://sqbu-github.cisco.com/huashi/OpenH264-Release2FTPServer>
- In wme, also using [curl](#) tools to download from ftp server

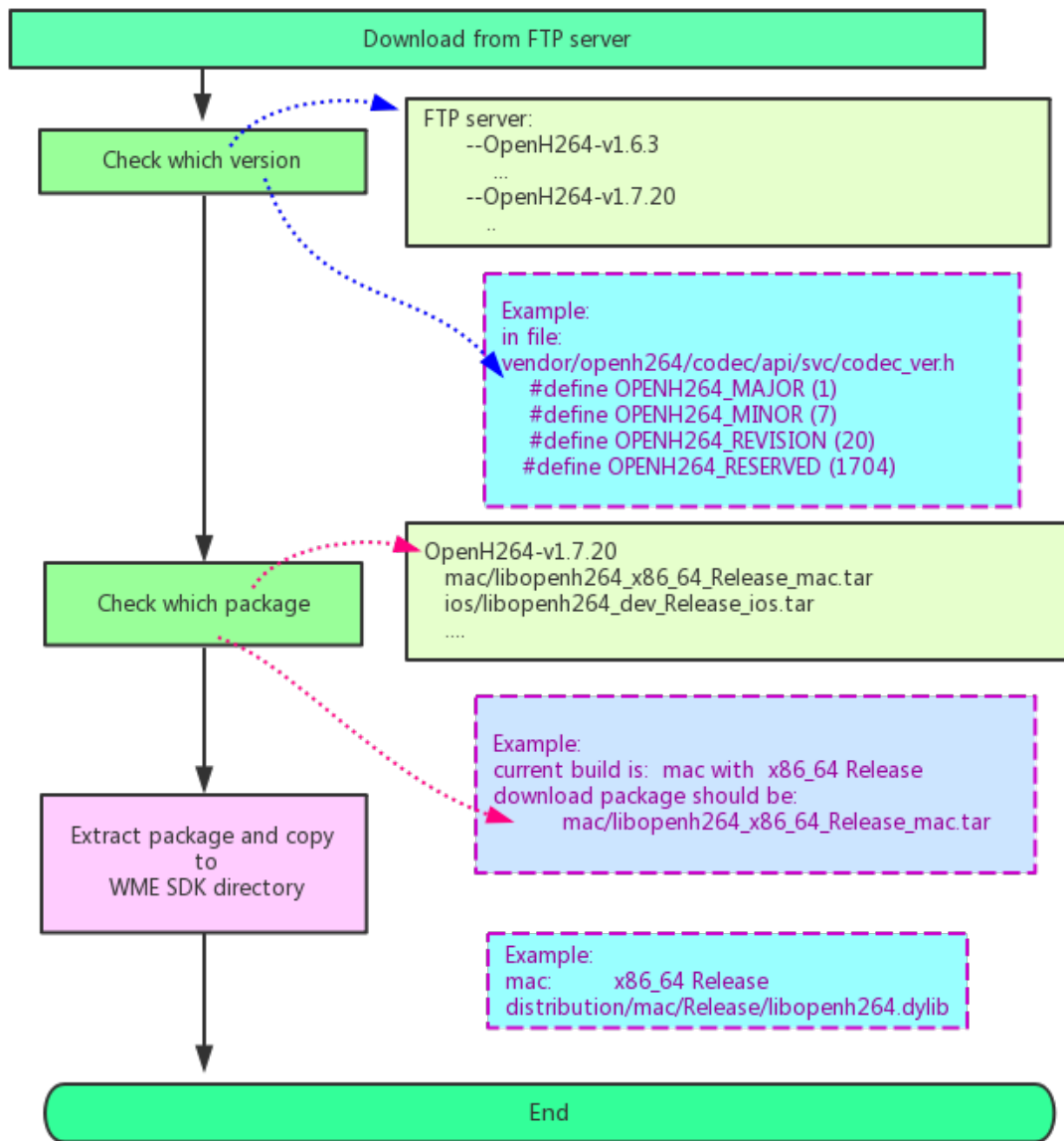
Index of /JenkinsRelease/

Name	Size	Date Modified
 [parent directory]		
 OpenH264-v1.6.3/		12/12/16, 1:49:00 PM
 OpenH264-v1.6.4/		12/17/16, 6:17:00 PM
 OpenH264-v1.6.5/		12/23/16, 10:15:00 AM
 OpenH264-v1.6.6/		1/19/17, 10:43:00 AM
 OpenH264-v1.6.7/		1/19/17, 10:08:00 AM
 OpenH264-v1.6.8/		3/1/17, 3:28:00 PM
 OpenH264-v1.7.0/		3/16/17, 3:39:00 PM
 OpenH264-v1.7.1/		3/29/17, 6:16:00 PM
 OpenH264-v1.7.20/		4/20/17, 9:47:00 AM

Name	Size	Date Modified
 [parent directory]		
 android/		3/16/17, 3:39:00 PM
 ios/		3/16/17, 3:35:00 PM
 linux/		3/16/17, 3:38:00 PM
 mac/		3/16/17, 3:37:00 PM
 win/		3/16/17, 3:36:00 PM

Name	Size	Date Modified
 [parent directory]		
 api_mac.tar	63.0 kB	3/16/17, 3:37:00 PM
 exefile_x86_64_Debug_mac.tar	8.1 MB	3/16/17, 3:37:00 PM
 exefile_x86_64_Release_mac.tar	6.1 MB	3/16/17, 3:37:00 PM
 exefile_x86_Debug_mac.tar	7.8 MB	3/16/17, 3:37:00 PM
 exefile_x86_Release_mac.tar	6.0 MB	3/16/17, 3:37:00 PM
 libgmpopenh264_x86_64_Debug_mac.tar	1.5 MB	3/16/17, 3:37:00 PM
 libgmpopenh264_x86_64_Release_mac.tar	1.1 MB	3/16/17, 3:37:00 PM
 libgmpopenh264_x86_Debug_mac.tar	1.4 MB	3/16/17, 3:37:00 PM
 libgmpopenh264_x86_Release_mac.tar	1.1 MB	3/16/17, 3:37:00 PM
 libopenh264_x86_64_Debug_mac.tar	1.3 MB	3/16/17, 3:37:00 PM
 libopenh264_x86_64_Release_mac.tar	1.1 MB	3/16/17, 3:37:00 PM
 libopenh264_x86_Debug_mac.tar	1.3 MB	3/16/17, 3:37:00 PM
 libopenh264_x86_Release_mac.tar	1.1 MB	3/16/17, 3:37:00 PM

2.1 Basic flowchart



2.2 Script

2.2.1 check version

```
build/ios/buildAndupdate_openh264.sh  
build/mac/buildAndupdate_openh264.sh  
build/windows/build_openh264.py
```

parse openh264 version info from:

vendor/openh264/codec/api/svc/codec_ver.h

```
#ifndef CODEC_VER_H  
#define CODEC_VER_H  
  
#include "codec_app_def.h"  
  
static const OpenH264Version g_stCodecVersion = {1, 7, 20, 1704};  
static const char* const g_strCodecVer = "OpenH264 version:1.7.20.1704";  
  
#define OPENH264_MAJOR (1)  
#define OPENH264_MINOR (7)  
#define OPENH264_REVISION (20)  
#define OPENH264_RESERVED (1704)  
  
#endif // CODEC_VER_H
```

Script for check version:

Example for android

mediaengine/shark/bld/client/android/build.py

```
def parse_openh264_version():  
    global package_version, abi_archs  
    openh264_version_file = os.path.join(openh264_dir, "codec/api/svc/codec_ver.h")  
    version_info = open(openh264_version_file).readlines()  
    for line in version_info:  
        if ( re.search("g_strCodecVer", line) ):  
            full_version = re.findall("\\d+", line)  
            package_version = full_version[1] + "." + full_version[2] + "." + full_version[3]
```

ios/mac, using tools from to complete all version check and download process

<https://sqbu-github.cisco.com/huashi/OpenH264-Release2FTPServer>

2.2.1 download

ios/mac, using tools from to complete all version check and download process

<https://sqbu-github.cisco.com/huashi/OpenH264-Release2FTPServer>

```
def download_openh264_lib():
    global package_version, package_url, openh264_lib
    tar_file = "libopenh264_" + abi_archs[app_abi] + "_Release_android.tar"
    package_url = package_url + "OpenH264-v" + package_version + "/android/" + tar_file
    download_cmd = "curl -u OpenH264:wme@cisco " + package_url + " -o " + tar_file
    print ("openh264 skip build and download from ftp server")
    print("download_cmd is: " + download_cmd)
    os.system(download_cmd)
    extract_cmd="tar -xvf " + tar_file
    os.system(extract_cmd)
    shutil.copy(openh264_lib, openh264_dir)
```

ios/mac, using tools from to complete all version check and download process

<https://sqbu-github.cisco.com/huashi/OpenH264-Release2FTPServer>

```
function DownloadOpenH264(){
    cd ${CODEC_PROJECT_PATH}
    OpenH264Dir=`pwd`

    #clean previous download history
    git clean -f ./*.tar
    rm -rf ${TargetDir}
    [ -d openh264-jenkins ] && rm -rf openh264-jenkins

    #clone download script
    ScriptCloneCmd="git clone git@sqbu-github.cisco.com:OpenH264/openh264-jenkins.git openh264-jenkins"
    echo "ScriptCloneCmd is ${ScriptCloneCmd}"
    ${ScriptCloneCmd}
    if [ $? -ne 0 ]; then
        echo "Error: clone download script from openh264-jenkins error"
        exit 1
    fi

    #always download the release openh264 library for WME
    bash openh264-jenkins/run_DownloadFromFTPServer.sh "ios" "lib" "${Arch}" "${Config}" "${OpenH264Dir}"
    if [ $? -ne 0 ]; then
        echo "Error: download openh264 ios lib from ftp server failed"
        exit 1
    fi

    tar -xvf libopenh264_${Arch}_${Config}_ios.tar
    rm -rf openh264-jenkins
    cd -
}
```

3. To do

- ✚ unify build and update script in openh264 internal repos
- ✚ windows build support VS2015, VS2017 etc.
- ✚ Incremental compile for all OS bout script level and Make file level