OnStream MediaPlayer+ Download Manager Implementation Guide
For Android

SDK Version 3.16.0

VisualOn, _Inc._

June, 2015

Version 1.3

20150630

Copyright/Confidentiality Notice

Table of Contents

1 About This Manual

1.1 Overview

This manual describes the OnStream® MediaPlayer+ (referred hereafter to as OSMP+) Download Manager with Android projects on the following topics:

- Features
- Compatibility
- DRM key management
- UI related operations
- Basic SDK integration
- Integration sample code

1.2 Audience

This manual is intended for developers who need to create a flexible and high-performance media player that supports streaming asset download.

Developers are assumed to be familiar with:
- Android SDK/ADT
- Eclipse IDE
- Java Native Interface (JNI)
- Android NDK
- Java, C, and C++ programming languages
- OSMP+ SDK

1.3 Revision History

Table 1 lists revisions made to this manual since the last version.

Table 1: Revision history

Revision	Product version	Date	Description
1.0	3.9	2013-12-18	Initial document
1.1	3.13.20	2014-11-25	Updated the Sample Code section
1.2	3.15.0	2015-04-10	Updated Chapter 2: Download Manager features, supported DRM engines, and DRM key management
1.3	3.16.0	2015-06-30	Updated Download Manager workflow diagram

1.4 Related Documents

The following documents (included with your installation package) provide additional information related to this manual:
- *API Reference Manual*

- _OnStream MediaPlayer+ User Guide_

1.5 Abbreviations

Table 2 defines the abbreviations used in this manual.

Table 2: Abbreviations

Abbreviation	Description
API	Application Programming Interface
DASH	Dynamic Adaptive Streaming over HTTP
DRM	Digital Rights Management
HLS	HTTP Live Streaming
IDE	Integrated Development Environment
JNI	Java Native Interface
SDK	Software Development Kit
VoD	Video on Demand

2 Overview

OnStream$^{\circledR}$ MediaPlayer+ (OSMP+) Download Manager feature enables downloading of Video-On-Demand (VOD) content for offline viewing. It supports VOD asset download using HTTP Live Streaming (HLS) and Smooth Streaming for playback on Android and iOS devices. License key management is available for supported DRMs to ensure playback of encrypted content in offline mode.

The OSMP+ player Download Manager supports downloading of VOD content to local file storage while playing with playback. Bitrate, audio track, and subtitle track can be selected for the content to be downloaded. Download of same content/bitrate can be resumed automatically from last break point. Content management such as start, stop, pause, delete, and progress status are available. Error messaging for download failure due to connection failure or full storage is also supported.

Download Manager downloads content to be stored at the local storage with the content trunk file (.ts) structure kept for offline playback. The OSMP+ player downloads the HLS or Smooth Streaming segments to the device storage, and then automatically generates a .manifest file (Smooth Streaming) and a .m3u8 file (HLS) for offline playback.
Note: The content and playlist can be downloaded without DRM key, however the downloaded source cannot be played if the DRM key is not available or expired.

The downloaded content is played as a local file, and user does not need create a local virtual server for hosting playback. Meanwhile, the OSMP+ SDK defines the local directory for storing the downloaded contents by setting the value of *LocalDir* in the *VOOSMPStreamingDownloader.VOOSMPStreamingDownloader.open(String source, int flag, String localDir)* API.

2.1 Features

The Download Manager module supports the following features:
- Download content to local storage while playing with playback or download only.
  Note: If Download Manager is turned on, the **Select Asset** dialog will prompt when starting the playback.
- Select video bitrate, audio track, and subtitle track to be downloaded.
  Note: If the Download Manager feature is turned on, you will be prompted to select only one video bitrate, audio track, or subtitle track for downloading in the **Select Asset** dialog.
- Check the DRM key expiration. See 2.3 DRM Key Management for more information about DRM key management.
- Report and display download progress.
- Resume download from the last break point.
- Report errors due to connection failure or full device storage.
- Start, stop, pause, resume, delete download from the application.

2.2 Compatibility

The download manager functionality is available with limitations:

- **Operating System (OS):** Android and iOS

- **Streaming protocol:** VoD for HLS and Smooth Streaming
- **DRM:** Verimatrix, Viaccess-Orca, Amazon, CableVision, Comcast, Conax, Intertrust, KKBox, LGUPlus, Nextscape, Arris, Showtime, and Tivo

Note: Although OSMP+ Download Manager supports the above DRMs, only Verimatrix provides the interface to download the DRM key to local device. That is, the downloaded DRM contents except Verimatrix cannot be played if the OSMP+ player cannot connect to their license servers, as the remaining DRM engines store the DRM key in the license server other than downloading the key to local device.

2.3 DRM Key Management

Verimatrix DRM supports downloading the key together with the content to local device when Download Manager is turned on. The OSMP+ player enables Verimatrix DRM module to download the Verimatirx DRM key to local device by default.

Note: The Verimatrix offline key mechanism only works for HLS on Android platform.

- If Verimatrix downloads the key to the device storage, the OSMP+ SDK invokes the *getDRMKeyExpiredStatus()* API to check the key for every playback, and user needs to manually access the HTTP URL again to update the local key once the key is expired.
- If user does not want to download Verimatrix key with the playback, do not invoke the *VOOSMPStreamingDownloader.setDRMVerificationInfo()* API to keep the DRM key in the server. Under this condition, the player will connect the license server to verify the DRM key when opening the downloaded content next time and the downloaded content cannot be played if the key verification fails.

2.4 UI Related Operations

2.4.1 Enabling Download Manager

To enable the Download Manager feature, select the checkbox next to **Download** from **Options**, as shown in Figure 1.
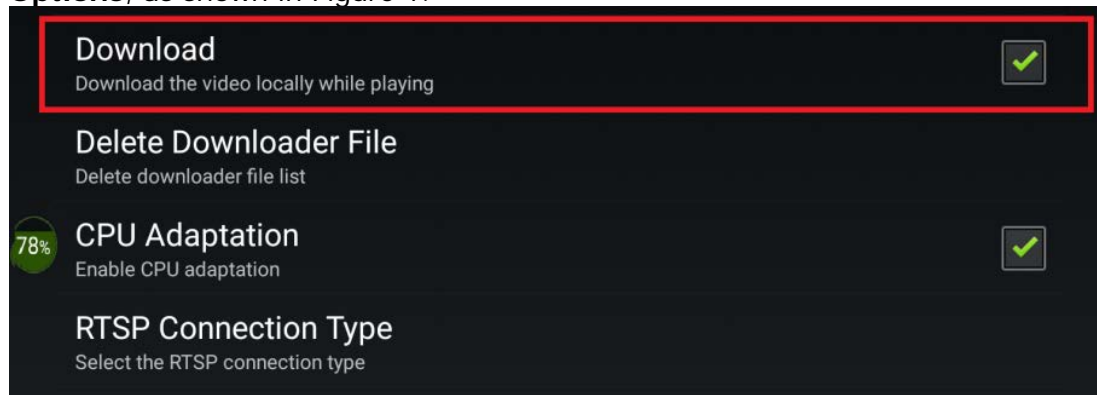

Figure 1: Enable Download Manager

2.4.2 Specifying assets

After turning on Download Manager, you will be prompted to specify the video bitrate, audio, and subtitle to be downloaded when starting the playback, as shown in Figure 2.

Note: The Select Asset dialog might vary from the stream that you selected to download,

for example, the Subtitle part will not appear if the selected stream does not contain any subtitle track.
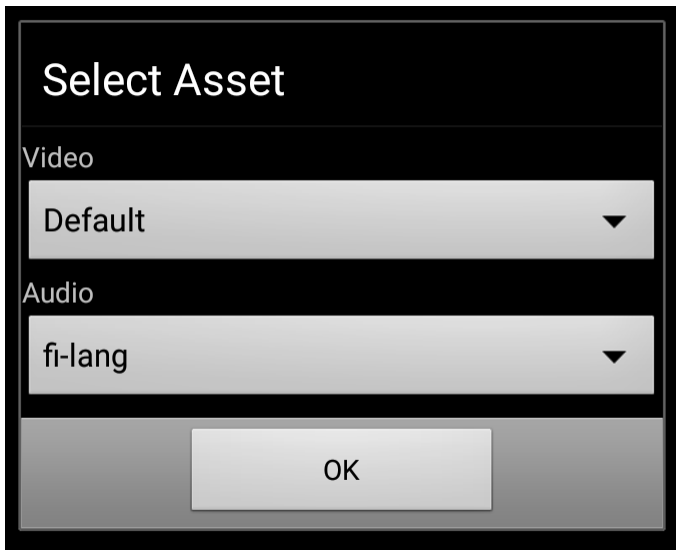

Figure 2: Select assets

2.4.3 Deleting downloaded content(s)

To delete the downloaded content(s):

1. Tap **Delete Downloader File** from **Options**.
2. Select the file(s) that you want to delete, and then tap **Delete**.

   Or

   Tap **Delete All** to remove all downloaded files from the device storage.
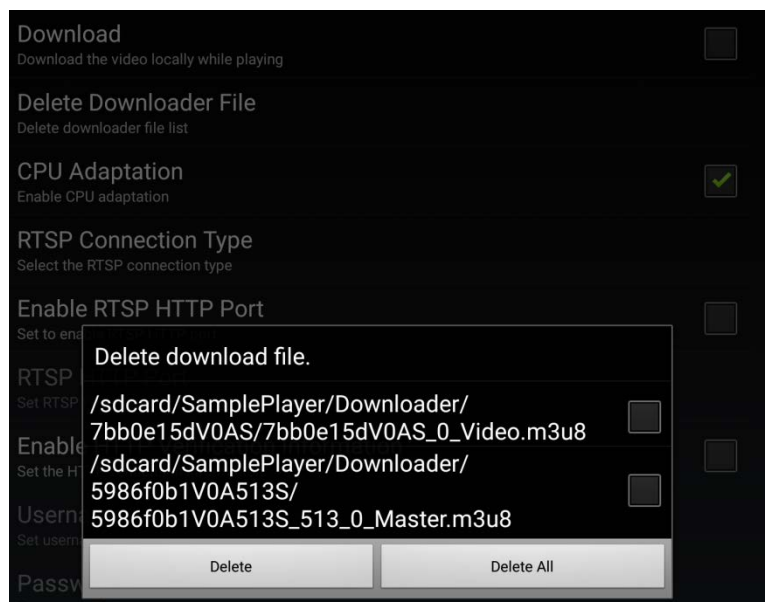

Figure 3: Delete downloaded files

3 Basic Integration

This section describes the Download Manager integration with the OSMP+ Android client. Figure 1 illustrates the basis workflow for integrating Download Manager with the OSMP+ SDK Android project.
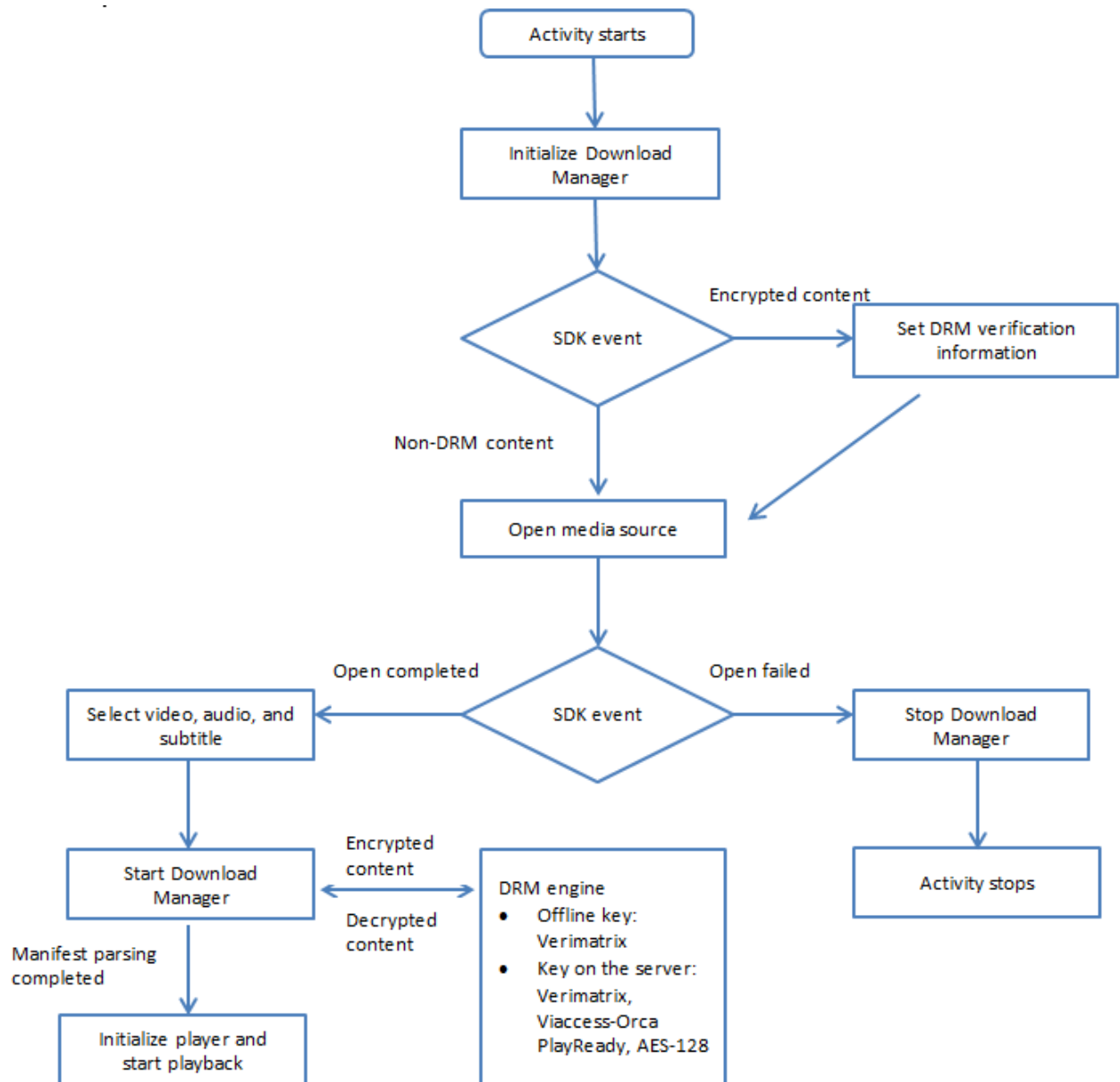

Figure 4: Diagram of Download Manager workflow

4 Integration Sample Code

4.1 Initializing Download Manager

See the following for the initialization sample code.

```
m_downloader = new VOOSMPStreamingDownloaderImpl();
String apkPath = CommonFunc.getApkPath(m_context);
VOOSMPStreamingDownloaderInitParam init = new
VOOSMPStreamingDownloaderInitParam();
init.setContext(m_context);
init.setLibraryPath(apkPath);
public VOOSMPStreamingDownloaderListener  downloaderListener = new
VOOSMPStreamingDownloaderListener() {
    @Override
    public VO_OSMP_RETURN_CODE
onVOStreamingDownloaderEvent(VO_OSMP_CB_STREAMING_DOWNLOADER_EVENT_ID
event, int arg1,int arg2, Object arg3) {
                ….
}
}
m_downloader.init(downloaderListener, init);
```

4.2 Opening Media Source

See the following for the sample code for opening the media source.

```
m_cDownloader.open( strVideoPath, 0, strLocalDownloadPath);
```

In the above sample, the *strVideoPath* is the link address that is to be downloaded and *strLocalDownloadPath* is the path that is added into the local files.

4.3 Handling Download Manager Event

See the following for the sample code for handling Download Manager Event.

```
public VO_OSMP_RETURN_CODE onVOStreamingDownloaderEvent(
            VO_OSMP_CB_STREAMING_DOWNLOADER_EVENT_ID event, int arg1,
int arg2,Object obj) {
    switch(event){
        case
VO_OSMP_CB_STREAMING_DOWNLOADER_SYNC_AUTHENTICATION_DRM_SERVER_INFO:
//Verimatrix DRM download support
        {
            VOOSMPVerificationInfo info = new VOOSMPVerificationInfo();
```

```
            info.setDataFlag(1);
            if(m_verificationString != null)
                info.setVerificationString(m_verificationString);
            m_downloader.setDRMVerificationInfo(info);
          break;
        }
   case VO_OSMP_CB_STREAMING_DOWNLOADER_OPEN_COMPLETE:
   //Select video/audio/subtitle track to download;
   m_downloader.selectVideo(index);
   m_downloader.commitSelection();
   …
   m_downloader.start();
   break;
   case VO_OSMP_CB_STREAMING_DOWNLOADER_MANIFEST_OK:
   //This is the download local path,it can be play throuth OSMP SDK now
   String localURL = String(obj);
   m_sdkPlayer.open(localURL……);
          break;
   …
   }
   }
```

4.4 Selecting Track

To select a track:

1.   Get the number of video tracks by using *int count = m_downloader.getVideoCount();*

2.   Use *m_downloader.selectVideo(index)* and *m_downloader.commit()* to select the corresponding index of track for playback.

   ▪   If the first track is selected, the index value is 0. Increase the index value in the ascending order. For example, if the second track is selected, the index value is 1.

   ▪   Set the index value to -1, which means to adjust the video track automatically according to the network bandwidth.

The Downloader Manager also extends the interface, so that it can handle track operations as what OSMP handles with tracks.

Note: For audio and subtitle selection, the index value cannot be set to -1. Refer to *VOCommonPlayerAssetSelection* for the all related functions.

4.5 Starting Download

See the following for the sample code for starting the download.
```
m_downloader.start();
```

4.6 Stopping Download

See the following for the sample code for stopping the download.
```
m_downloader.stop();

m_downloader.close();
```