

# NASA Planetary Data System (PDS) and MP4 (H.264 and AAC) Video

Prepared for the Planetary Data System by Trent Hare, [thare@usgs.gov](mailto:thare@usgs.gov), May 2017, updates by Lev Nagdimunov and Lyle Huber

---

Introduction to MP4 (MPEG-4 Part 14) .....	1
H.264 video codec (MPEG-4 Part 10): .....	1
Advanced Audio Coding (AAC) audio codec (MPEG-4 Part 3): .....	1
Recommended viewing application .....	2
Recommended method to convert to MP4 (H.264 and AAC) .....	2
Quality Discussion .....	3

---

## Introduction to MP4 (MPEG-4 Part 14)

The PDS has recently approved a restricted set of digital video as a valid “*document*” format. PDS *documents*, which intended to assist in understanding data contained in the PDS archive, mainly include textual or text-based information, but may also include graphics and/or images. These videos are not intended to archive scientific data for distribution or analysis, but rather to facilitate the understanding of the aforementioned scientific data. This could include, but is not limited to, an animated video showing 3D rendered scenes of orbital coverage over the time of the data set or an animation of data graphed over time. Videos may not supplant text-based documentation, which must continue to include everything necessary to understand the uncompressed scientific data being archived.

For the video format, as of 2017, the PDS has chosen the MP4 video format (MPEG-4 Part 14) encoded with the H.264 (MPEG-4 Part 10) and AAC codecs (MPEG-4 Part 3). MP4 is a standard developed by the Moving Picture Experts Group (MPEG) which is commonly used for sharing video files on the Internet. However, MP4 (generally with a file extension of .mp4), is simply a container format which does not specify the type of video and/or audio codec (COmpression/DECompression) method used within the format. For MP4 video, the PDS requires, due to its broad support, (1) the video codec H.264. For MP4 audio, the PDS requires (2) the Advanced Audio Coding (AAC). Both of these codecs, H.264 and AAC are lossy formats, and are not suitable for archiving scientific data.

*H.264 video codec (MPEG-4 Part 10):* While H.264 is patent-encumbered and licensed by MPEG LA, Cisco has released open source H.264 implementation called OpenH264. This release enables any open source project to incorporate Cisco’s H.264 module without paying MPEG LA license fees. OpenH264 is released under the terms of the Simplified BSD License (URL: <http://www.openh264.org/>). As an alternative to the Cisco implementation, another open source software library and application, called x264, also exists for encoding and decoding video streams into the H.264 compression format. X264 is released under the terms of the GNU GPL (URL: <http://www.videolan.org/developers/x264.html>). Lastly, the PDS, without restriction, can host H.264 encoded video since it is free to distribute for non-commercial use.

*Advanced Audio Coding (AAC) audio codec (MPEG-4 Part 3):* AAC was developed by several companies including AT&T Bell Laboratories, Fraunhofer IIS, Dolby Laboratories, Sony Corporation and Nokia. It is specified both as Part 7 of the MPEG-2 standard and as Subpart 4 in Part 3 of the MPEG-4 standard. While declared an international standard in 1997, it was made popular in the mid-2000’s by Apple’s iTunes application (and portable music devices) and included in videos on YouTube, now owned by Google. License fees are only due on the sale of encoders and/or decoders applications. No licenses are

required for the PDS to stream or distribute content in AAC format. AAC encoders released by the FFmpeg team are LGPL-licensed and open source (URL: <https://ffmpeg.org/>). For music- or audio-only files, the .m4a extension is generally used (for example audio-only conversions see: <https://trac.ffmpeg.org/wiki/Encode/AAC>).

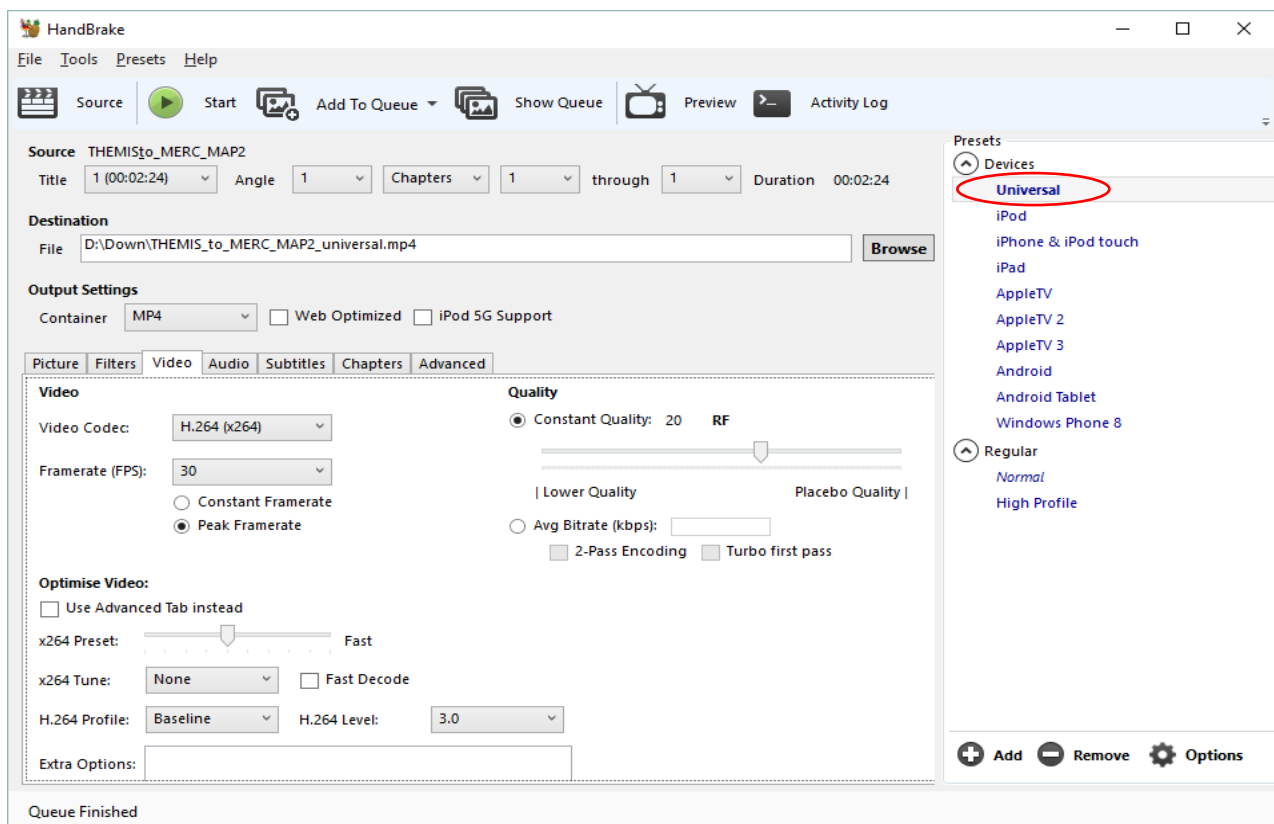
### Recommended viewing application

There are many different applications to view MP4 videos encoded with the H.264 and AAC codecs. One open source application, with extremely broad support across a large variety of operating systems (including portable devices and phones), is the VLC media player by VideoLAN (URL: <https://www.videolan.org/vlc/>).

### Recommended method to convert to MP4 (H.264 and AAC)

Nearly all video production software will likely support H.264 video and AAC audio encoding during the creation of a MP4 formatted file.

1. If your video application does not support H.264 and/or AAC encoding, the open source program HandBrake can help to convert from most any video format to MP4 (URL: <https://handbrake.fr/>). Below is a screen shot of Handbrake with a video loaded for conversion. Simply choosing “Universal” under devices (circled in red) should provide you with recommended settings for a compliant PDS video document.



2. Another open source method to convert from a different video types to MP4 (using H.264 and AAC encoders) is *ffmpeg*, freely released by the FFmpeg team (URL: <https://ffmpeg.org/>). This routine can also convert a series of image frames (e.g. JPEGs or PNGs) to an MP4 video. For command-line options see URL: <https://trac.ffmpeg.org/wiki/Encode/H.264>.

## Recommended method to test for MP4 codecs (to help meet PDS compliance)

There are several methods to test a MP4 video for compliance with H.264 video and AAC audio encoded streams. Two simple methods are described below.

1. The first method is to use HandBrake above. Simply load the video using the “Source” button. Under the activity log (scan tab), HandBrake will list the MP4 file metadata for the used codecs.
2. Another open source and command-line routine, called *ffprobe*, freely released by the FFmpeg team and built across several operating systems, can also be used (URL: <https://ffmpeg.org/>). Here is an example run. Note the two lines which are highlighted in bold (lines removed for brevity).

```
$ ffprobe THEMIS_to_MERC_MAP2_universal.mp4
ffprobe version N-86129-g1e8daf3 Copyright (c) 2007-2017 the FFmpeg developers
. . .
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from 'THEMIS_to_MERC_MAP2_universal.mp4':
  Metadata:
    major_brand      : mp42
    minor_version    : 512
    compatible_brands: isomiso2avc1mp41
    creation_time    : 2017-05-23T18:47:55.000000Z
    encoder          : HandBrake 0.10.5 2016021100
  Duration: 00:02:24.82, start: 0.000000, bitrate: 429 kb/s
    Stream #0:0(und): Video: h264 (Constrained Baseline) (avc1 /
0x31637661), yuv420p(tv, bt709), 1280x720 [SAR 52243:52200 DAR 587:435],
262 kb/s, SAR 26729:26707 DAR 587:435, 15 fps, 15 tbr, 90k tbn, 180k tbc
    Metadata:
      handler_name    : VideoHandler
    Stream #0:1(eng): Audio: aac (LC) (mp4a / 0x6134706D), 44100 Hz,
stereo, fltp, 162 kb/s (default)
    Metadata:
      handler_name    : Stereo
```

## Quality Discussion

The “Universal” present in HandBrake defaults the video quality setting to 20 “Constant Quality” and will define a variable frame rate with the allowable peak of 30 frames per second. Using these defaults will result in a quality video conversion suitable for archival. However, if you are concerned about the video quality, set the HandBrake quality to **16** (which will increase the video bitrate) and set the “Optimize Video” x264 Present to **Slower**. The default bitrate for the AAC audio encoder will be set to a 160 kilobytes per second, which is more than sufficient for speech and most music. If you are also concerned about the audio quality, then set the bitrate to **320** kilobytes per second. Note that changing these default settings could result in a much larger file size.

Lastly, for the video size and aspect ratio, this is generally defined by the original application or camera, but current recommended sizes include:

- 720p: 1280x720 (HD, High Definition TV quality), current recommended size for web
  - 1080p: 1920x1080 (Full HD, Blu-ray quality), **current recommended size for archival and web**
  - 2160p: 3840x2160 (Ultra-HD, 4K quality), only beginning to grow in popularity
- where p stands for progressive scan and width x height are in pixels. All are 16:9 ratio, which is common for high definition (HD) televisions and web services like YouTube.