

Harnessing MPEG-2 Transport Stream with LEADTOOLS

Introduction

Like many booming industries, audio and video streaming is supported by a large cast of technologies serving in a wide variety of roles from high-end military drones to home entertainment and everything in between. The cast takes the form of audio and video compression algorithms within codecs, multiplexers, demultiplexers and the applications that bring them to life. Of these, MPEG-2 Transport Stream has emerged as one of the most versatile and widely used video streaming formats.

When it comes to streaming real-time multimedia content in high-stakes scenarios such as Unmanned Aerial Vehicles (UAVs), surveillance and security you need something with the utmost reliability since the lives and well-beings of others are at risk. This white paper will dive into the many advantages and uses for MPEG-2 Transport Stream and how to incorporate it into your application with LEADTOOLS Multimedia SDKs.

Benefits and Uses of MPEG-2 Transport Stream

MPEG-2 Transport Stream (MPEG-2 TS) is like the drummer of your favorite rock band. You probably don't know his name and wouldn't recognize him if he was standing right next to you. Without him though, the music you love would be far less enjoyable and potentially not even exist at all. MPEG-2 Transport Stream is at the heart of many multimedia technologies you benefit from every day, such as Blu-ray, video-on-demand services, security and traffic monitoring systems, reconnaissance drones and more, providing you with entertainment and keeping you safe and secure as you go about your life. This widespread adoption didn't happen by accident. It was gained through a well-designed specification that facilitates audio and video streaming across a huge gamut of mediums and scenarios.

MPEG-2 Transport Stream is a flexible container format which can employ several video formats including MPEG-2, MPEG-4 or h.264 video combined with AAC, AC3 or MPEG-1 audio. Each format has its own advantages, but the most popular solution is h.264 video with AAC audio. This combination offers the best in both quality and bandwidth, making it an ideal choice for situations involving real-time, over-the-air streaming or slower connection speeds using either TCP or UDP protocols.

There are other considerations when choosing your audio and video codecs as well. Interframe compressed formats such as MPEG-4 and MPEG-2 perform better with higher bit rates and bandwidths and are ideal for recording and playback via disc, DVR or live television.

Finally, one of the most important characteristics of MPEG-2 Transport Stream is the ability to embed non-destructive Key-Length-Value (KLV) metadata. Unlike its destructive counterparts which permanently burn time stamps, coordinates and other information into the video data, KLV can be used to store a large amount of textual metadata in a very compact and live-streaming efficient manner. The viewing application can display as much or as little metadata as desired, which is perfect for implementing closed captioning or customized players. It is also more efficient than destructive metadata because the binary encoding format requires less CPU operations to write than most image processing routines that would burn the text into the image.

MPEG-2 Transport Stream with LEADTOOLS

LEADTOOLS takes normally complex DirectShow and Media Foundation projects and turns them into simple tasks with its high level SDKs for .NET and C/C++. Each of the primary MPEG-2 Transport Stream related activities including sending, receiving and working with KLV metadata can all be done with only a few lines of code.

Streaming MPEG-2 Transport with KLV Metadata

All of the core components of a multimedia application (capture, conversion and playback) are bundled into easy to use, high-level .NET objects (shown in the following examples) or COM objects. Whether you are capturing live video and audio with the [CaptureCtrl](#), or converting a file that already exists with the [ConvertCtrl](#), outputting your results to disk or a live stream follows the same steps once your source is selected.

Instead of manually building a DirectShow Graph or Media Foundation Topology, all you need to do with LEADTOOLS is set your source, destination and desired format and LEADTOOLS will do all of the grunt work behind the scenes for you. You can also get instances of any filter along the way to fine-tune your settings for quality, compression, bit rate padding, TV broadcast specification compliance and more.

```
// Set up the Convert Control
_convertCtrl.SourceFile = srcFile;
_convertCtrl.AllowedStreams = StreamFormatType.AudioVideoCC;
_convertCtrl.PreferredMPEG2Splitter =
    Constants.Filter_MPEG2_Transport_Demux;
_convertCtrl.TargetFormat = TargetFormatType.MPEG2Transport;
_convertCtrl.TargetFile = @"udp://127.0.0.1:9005";

// Get the MPEG-2 Multiplexer from the graph and change some settings
LMMpg2MxT mux = _convertCtrl.GetSubObject(ConvertObject.TargetFilter)
    as LMMpg2MxT;
if (mux != null)
{
    mux.ResyncInterval = 10;
    Marshal.ReleaseComObject(mux);
}

// Start converting to stream
_convertCtrl.StartConvert();
```

LEADTOOLS comes with everything you need to create and parse any KLV data. You have full control and can therefore make your own custom implementation or adhere to established standards such as the various UAV Metadata Sets established by the Motion Imagery Standards Board (MISB).

```

// write a timestamp key
double longitude = -80.8433333;
double latitude = 35.2269444;
string platform = "LEAD Demo";

// clear out any existing data
m_pMpgMux.KlvBuilder.Clear();

// write the UDS timestamp
m_pMpgMux.KlvBuilder.InsertUInt64(-1,
    "06 0E 2B 34 01 01 01 03 07 02 01 01 01 05 00 00", (ulong)timestamp);

// write the UDS sensor latitude
m_pMpgMux.KlvBuilder.InsertDouble(-1,
    "06 0E 2B 34 01 01 01 03 07 01 02 01 02 04 02 00", latitude);

// write the UDS sensor longitude
m_pMpgMux.KlvBuilder.InsertDouble(-1,
    "06 0E 2B 34 01 01 01 03 07 01 02 01 02 06 02 00", longitude);

// write the UDS platform designation
m_pMpgMux.KlvBuilder.InsertString(-1,
    "06 0E 2B 34 01 01 01 01 01 01 20 01 00 00 00 00", platform);
LMMpg2MxTLib.ILMKlvBuilder group;
group = m_pMpgMux.KlvBuilder.CreateGroup(
    "06 0E 2B 34 02 0B 01 01 0E 01 03 01 01 00 00 00");

// write the LDS timestamp
group.InsertLocalUInt64(-1, 2, (ulong)timestamp);

// write the LDS sensor latitude
group.InsertLocalInt32(-1, 13,
    (int)Math.Floor(0xFFFFFFFFE / 180.0 * latitude + 0.5));

// write the LDS sensor longitude
group.InsertLocalInt32(-1, 14,
    (int)Math.Floor(0xFFFFFFFFE / 360.0 * longitude + 0.5));

// write the LDS platform designation
group.InsertLocalString(-1, 10, platform);

// write the LDS checksum
group.AppendLocalChecksum(1, null);

// insert the UAS Datalink Local Metadata Set into the main builder
m_pMpgMux.KlvBuilder.InsertGroup(-1, null, group);

// get the full builder data and write it to the output
m_pMpgMux.WritePrivateData(
    (int)LMMpg2MxTLib.Mpg2MxT_WriteFlags.Mpg2MxT_WriteFlag_PTSTValid |
    (int)LMMpg2MxTLib.Mpg2MxT_WriteFlags.Mpg2MxT_WriteFlag_PTSTInSeconds,
    (double)elapsed / 1000.0, m_pMpgMux.KlvBuilder.GetData(), -1);

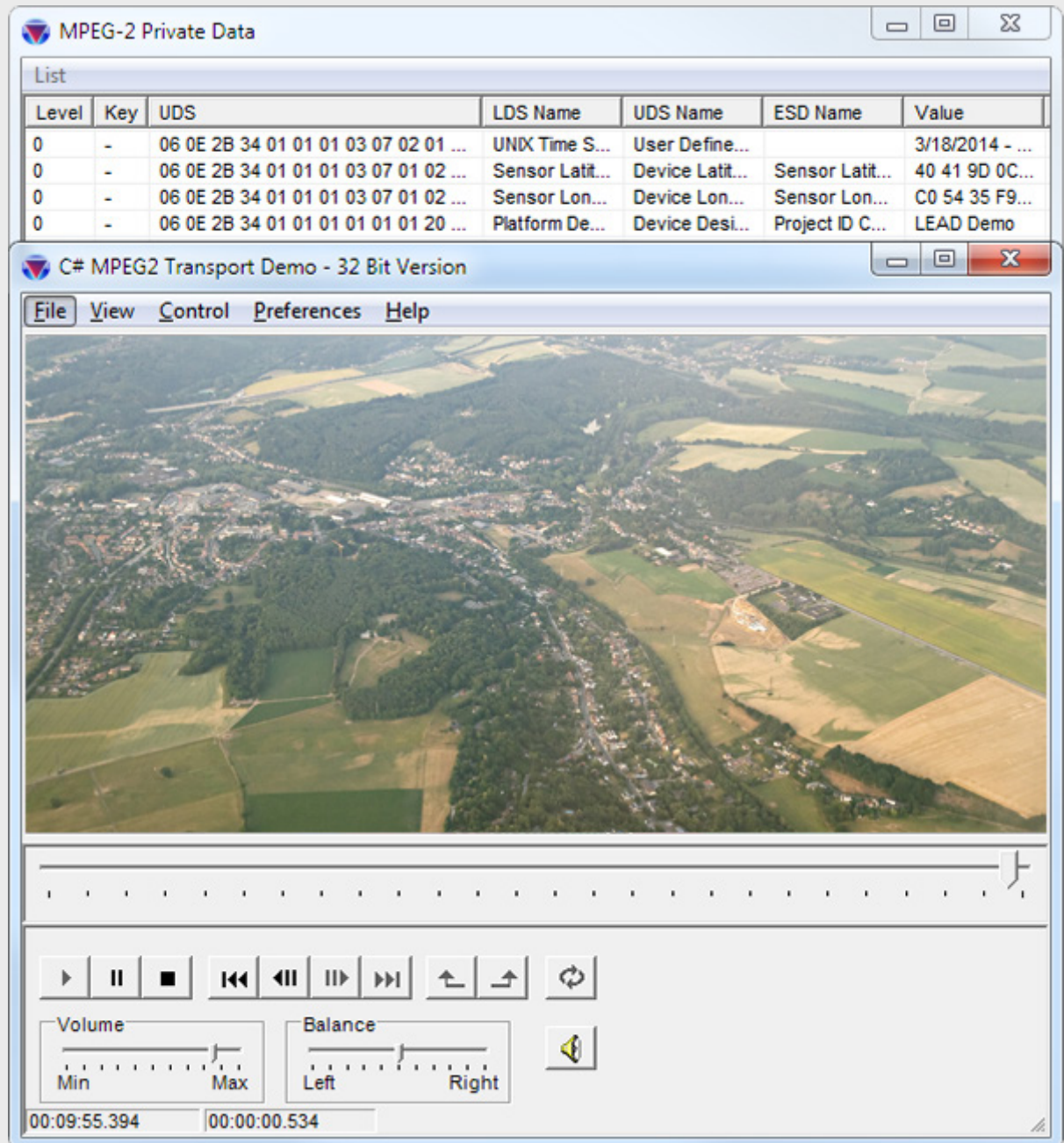
// close the stream
m_pMpgMux.ClosePrivateData();

```

Connecting to and Playing a Live MPEG-2 Transport Stream

Similarly, audio and video playback with the `PlayCtrl` is designed with the programmer in mind, requiring very little code to set up and play files or live streams while still providing low-level access and control for more advanced, custom-tailored applications. In most scenarios, playing a file or stream is as simple as setting the `SourceFile` property:

```
_playctrl.SourceFile = "udp://127.0.0.1:9005?Timeout=16";
```



Additional MPEG-2 Transport Stream Features in LEADTOOLS

The power of LEADTOOLS isn't limited to capture, conversion, playback and streaming. There is much more that easily integrates into your MPEG-2 Transport Stream application for a fully-featured and well-rounded product. A wide variety of audio and video processing filters including motion detection, video stabilizer, deinterlace, frame rate controller and more can be added to your stream or file. Additionally, full Digital Video Recorder (DVR) functionality complete with pause, rewind, jump to live and the like help elevate your entertainment, security or surveillance application to a whole new level.

All of this combined with the MPEG2-Transport Stream features discussed above come together to make LEADTOOLS the most dependable, flexible and programmer-friendly Multimedia SDK on the market.

Conclusion

This is just one of many real world solutions you can tackle with LEADTOOLS. Its state of the art MPEG-2 Transport Stream SDK is the most flexible and powerful product in its class, and LEADTOOLS offers an incredible value with its comprehensive family of toolkits for raster, document, medical and multimedia imaging. For more information on how LEAD Technologies can image-enable your application and boost your ROI, visit www.leadtools.com to download a free evaluation, or give us a call at +1-704-332-5532.

Sales: +1(704) 332-5532
sales@leadtools.com

Support: +1(704) 372-9681
support@leadtools.com



LEAD Technologies, Inc.
1927 South Tryon Street
Suite 200
Charlotte, NC 28203

About LEAD Technologies

With a rich history of over 25 years, LEAD has established itself as the world's leading provider of software development toolkits for document, medical, multimedia, raster and vector imaging. LEAD's flagship product, LEADTOOLS, holds the top position in every major country throughout the world and boasts a healthy, diverse customer base and strong list of corporate partners including some of the largest and most influential organizations from around the globe.

LEADTOOLS[®]
THE WORLD LEADER IN IMAGING SDKs

