This is a plugin based on the LZ4 library released under the BSD license (see above).

The scope of this library is to compress/decompress LZ4 archives and buffers on Android, iOS, tvOS*, Windows, OSX, Linux & webGL*.

The examples require some files that reside in the StreamingAssets folder. They are there just for testing purposes. In your final projects make sure you delete them so that you don't increase your build size.

The ios libraries are compiled as universal and bitcode enabled. That means that they will support 32 and 64 bit builds.
(non-bitcode enabled iOS plugins are provided. If you are creating a non bitcode enabled project please use these provided plugins!)

OSX bundle is compiled now as 64-bit only, since Apple store requires it.
The Windows and Linux libraries are compiled for x86 and x86_64 build modes.
The Android lib is compiled for armeabi-v7a, x86 and arm64-v8a.

***webGL/tvOS support compressing/decompressing of LZ4 buffers only.**

FEATURES:

Fast LZ4 compression and decompression with a clean and simple interface. Very easy to use.

- compress a file into a LZ4 file format.

- decompress an LZ4 file.

- compress a buffer into the LZ4 format.
- decompress an LZ4 compressed buffer into a buffer.

(The plugin provides a solution to store the uncompressed size of a compressed buffer in its footer.)

- Linux, iOS, Android, MacOSX can treat buffers as files. That means if you have a file in www.bytes you can perform operations directly on the buffer.
  For Android this is very useful since you can decompress from Streaming Assets without copying to Persistent data path.

!!! If you want to use only the lz4 plugin, please delete all the other plugins in their respective folders or use the single packages from the _plugin_packages folder!!!.

INSTRUCTIONS:

If you want to run a small example, compile and run the testScene.
It will download a small tif file and it will perform all the functions the lib provides.

See the LZ4.cs file for more comments and error codes.

In your project include in the Plugins folder the plugins you want to use and the LZ4.cs file and call the appropriate functions
as described below and shown in the demo scene.

FUNCTIONS:

---

float compress(string inFile, string outFile, int level, float[] progress);

Compress a file to LZ4.
Returns the rate of compression.
Full paths to the files should be provided.

inFile : The input file to compress.
outFile : The output compressed file.
level : level of compression (1 - 9).
progress : provide a single item float array to get the progress of the compression in real time. (only when called from a thread/task)

---

int decompress(string inFile, string outFile, ulong[] bytes, byte[] FileBuffer = null);

Decompress an LZ4 file.
Full paths to the files should be provided.
Returns 0 on success.

inFile : The input compressed file.
outFile : The output decompressed file.

Bytes : provide a single item ulong array to get the bytes currently decompressed in real time. (only when called from a thread/task)
FileBuffer : A buffer that holds an LZ4 file. When assigned the function will decompress from this buffer and will ignore the filePath. (Linux, iOS, Android, MacOSX)

---

bool compressBuffer(byte[] inBuffer, ref byte[] outBuffer, bool includeSize = true);

Compress a byte buffer in LZ4 format.
Returns true on success.

inBuffer : the uncompressed buffer.
outBuffer : a referenced buffer that will be resized to fit the lz4 compressed data.
includeSize : include the uncompressed size of the buffer in the resulted compressed one because lz4 does not include this.

---

byte[] compressBuffer(byte[] inBuffer, bool includeSize = true);

Compress a byte buffer in LZ4 format ad return a new buffer compressed.
Returns a new buffer with the compressed data.

inBuffer : the uncompressed buffer.
includeSize : include the uncompressed size of the buffer in the resulted compressed one because lz4 does not include this.

---

bool decompressBuffer(byte[] inBuffer, ref byte[] outBuffer, bool useFooter = true, int customLength = 0);

Decompress an lz4 compressed buffer to a referenced buffer.
Returns true on success.

inBuffer : the lz4 compressed buffer
outBuffer : a referenced buffer that will be resized to store the uncompressed data.
useFooter : if the input Buffer has the uncompressed size info.
customLength : provide the uncompressed size of the compressed buffer. Not needed if the useFooter is used!

---

int compressBufferPartialFixed (byte[] inBuffer, ref byte[] outBuffer,int partialIndex, int level, bool includeSize = true);

Compress a byte buffer in LZ4 format at a specific position of a fixed size outBuffer.
Returns compressed size (+4 bytes if footer is used).

inBuffer : the uncompressed buffer.
outBuffer : a referenced buffer of fixed size that could have already some lz4 compressed buffers stored.
partialIndex : the position at which the compressed data will be written to.
level : level of compression (1 - 9).
includeSize : include the uncompressed size of the buffer in the resulted compressed one because lz4 does not include this.

---

byte[] decompressBuffer(byte[] inBuffer, bool useFooter = true, int customLength = 0);

Decompress an lz4 compressed buffer to a new buffer.

Returns a new buffer with the uncompressed data.

inBuffer : the lz4 compressed buffer.
useFooter : if the input Buffer has the uncompressed size info.
customLength : provide the uncompressed size of the compressed buffer. Not needed if the useFooter is used!

---

int decompressBufferFixed(byte[] inBuffer, ref byte[] outBuffer, bool safe = true, bool useFooter = true, int customLength = 0);

Decompress an lz4 compressed buffer to a referenced fixed size buffer.
Returns the uncompressed size.

inBuffer : the lz4 compressed buffer
outBuffer : a referenced fixed size buffer where the data will get decompressed.
usefooter : if the input Buffer has the uncompressed size info.
customLength : provide the uncompressed size of the compressed buffer. Not needed if the usefooter is used!

This function is useful if you want to avoid memory allocations caused by new buffers or buffer resizing.

---

int decompressBufferPartialFixed (byte[] inBuffer, ref byte[] outBuffer, int partialIndex , int compressedBufferSize, bool safe = true, bool useFooter = true, int customLength = 0);

A function to decompress a buffer stored in a fixed size buffer that stores many compressed lz4 buffers.
Returns the uncompressed size,

inBuffer : input buffer that stores multiple lz4 compressed buffers.
outBuffer : a referenced fixed size buffer where the data will get decompressed.
partialIndex : position of an lz4 compressed buffer in the inBuffer.
compressedBufferSize : compressed size of the buffer to be decompressed.
safe : check if the uncompressed size is bigger then the size of the fixed buffer.
useFooter : if the input Buffer has the uncompressed size info.
customLength : provide the uncompressed size of the compressed buffer. Not needed if the usefooter is used!

[Android, iOS, Linux, MacOSX only]
int setFilePermissions(string filePath, string _user, string _group, string _other);

Sets permissions of a file in user, group, other.

Each string should contain any or all chars of "rwx".

Returns 0 on success.

SUPPORT:

For any questions, problems and suggestions please use this email address: elias_t@yahoo.com

forum: http://forum.unity3d.com/threads/7zip-lzma-and-zip-native-multiplatform-plugins.211273/