Subject: How to Zip cross-platform from IDL?
Posted by Brian Daniel on Fri, 22 Feb 2013 13:40:16 GMT
View Forum Message <> Reply to Message

I have an extensive IDL/ENVI program that writes a KML file output. The code writes image chips to files, points the KML to those files, creates info-balloons with all sorts of diagnostic information for each chip. This all works well. However, when I distribute the results, the KML and all the supporting images need to go with it. This is cumbersome, but solvable! I just need to ZIP it up into a KMZ file.

The catch: I need to be able to do this on Mac, Win and Linux. Linux and Mac can utilize the gzip functionality through the SPAWN command, but on Windows... I'm stumped. I can't assume I can install 7zip or WinZip and call those independently. Is there a way IDL can do this natively, or am I at the mercy of Windows standard calls?

Thanks for your comments.

-B

Subject: Re: How to Zip cross-platform from IDL?
Posted by Lajos Foldy on Tue, 16 Apr 2013 08:15:56 GMT
View Forum Message <> Reply to Message

Hi Chris,

On Tuesday, April 16, 2013 6:50:04 AM UTC+2, Chris Torrence wrote:
> Hi all,
>
>
>
> **Spoiler alert**
>
>
>
> Here's what we've got for IDL 8.2.3:
>
>
>
> FILE_TAR: Input files or directories, output to a tar file or to a memory buffer. Optional GZIP compression. Optional keyword to just get a list of files but don't do any real work.
>
>
>
> FILE_UNTAR: Input a tar file or a memory buffer, output all the files/directories. Automatically handles GZIP compression. Optional keyword to just get a list of files but don't do any real work.
>
>

>
> FILE_ZIP: Input files or directories, output to a zip file. Optional keyword to just get a list of files but don't do any real work.
>
>
>
> FILE_UNZIP: Input a zip file, output all the files/directories. Optional keyword to just get a list of files but don't do any real work.
>
>
>
> FILE_GZIP: Input file or files, output each to either gzip file or to a memory buffer.
>
>
>
> FILE_GUNZIP: Input gzip file or files, output the uncompressed files or to a memory buffer.
>
>
>
> ZLIB_COMPRESS: Input an array of any IDL numeric type, output an array with Deflate compression (with either no header, a ZLIB header, or a GZIP header).
>
>
>
> ZLIB_UNCOMPRESS: Input a byte array with compressed data, output an IDL numeric type (given the appropriate type and dimensions).
>
>
>
> Note that for all of the FILE* routines, the assumption is that you have files at one end or the other (or both) - you cannot go straight from data to a compressed memory buffer, and then go back to uncompressed data. Instead, if you want to do that, you can just use the raw ZLIB_COMPRESS/UNCOMPRESS routines.
>
>
>
> Regarding speed tests, I can TAR and UNTAR about 2000 small files in ~10 seconds on pokey Windows NTFS. On Linux it take 0.45 seconds.
>
>
>
> Using ZLIB_COMPRESS/UNCOMPRESS, it takes about 3 seconds to compress/uncompress ~100 MB of data.
>
>
>
> Cheers,
>

> Chris
>
> ExelisVIS

May I suggest to add LZO, bzip2 and xz as an option? With these, one would be able to choose a good trade-off between speed and compression ratio, eg. LZO (fastest) for developing code and xz (best compression ratio) for release mode.

regards,
Lajos

Subject: Re: How to Zip cross-platform from IDL?
Posted by Craig Markwardt on Tue, 16 Apr 2013 15:08:32 GMT
View Forum Message <> Reply to Message

On Tuesday, April 16, 2013 12:50:04 AM UTC-4, Chris Torrence wrote:
> Hi all,
>
>
>
> **Spoiler alert**

Those are nice additions. Can you do streaming un/compression? Or do you need to be able to store the entire file in memory or disk?

Craig

Subject: Re: How to Zip cross-platform from IDL?
Posted by markb77 on Tue, 16 Apr 2013 15:31:59 GMT
View Forum Message <> Reply to Message

On Apr 16, 11:08 am, Craig Markwardt <craig.markwa...@gmail.com>
wrote:
> On Tuesday, April 16, 2013 12:50:04 AM UTC-4, Chris Torrence wrote:
>> Hi all,
>
>> **Spoiler alert**
>

Great!!

> Those are nice additions. Can you do streaming un/compression? Or do you need to be able to store the entire file in memory or disk?

>
> Craig

I second this request for streaming un/compression capabilities

Mark

Subject: Re: How to Zip cross-platform from IDL?
Posted by chris_torrence@NOSPAM on Thu, 18 Apr 2013 13:40:08 GMT
View Forum Message <> Reply to Message

Hi Craig & Mark,

Right now it doesn't support streaming. I didn't want to bloat up the number of keywords & arguments to support streaming, and it would also make the documentation very confusing.

What exactly do you want to do with streaming? The problem with doing streamed compress/uncompress is that you need to chunk up the data in somewhat arbitrary ways. In particular, when doing uncompression, the uncompressor has to wait until it has gotten enough data to uncompress a particular chunk, while with compression, you might call it several times before it actually returns a chunk of compressed data. Very messy, especially for a vector-based language like IDL.

Is it possible that you could simply chunk up your data yourself, say in 1KB or 1MB chunks, compress those using the new zlib_compress(), send them "across", and then uncompress them? Even though you're doing the chunking yourself, I would bet that it would be simpler to code that up than to use any sort of streaming functions that we might provide.

Thoughts?

-Chris
ExelisVIS

Subject: Re: How to Zip cross-platform from IDL?
Posted by markb77 on Thu, 18 Apr 2013 15:58:41 GMT
View Forum Message <> Reply to Message

On Apr 18, 9:40 am, Chris Torrence <gorth...@gmail.com> wrote:
> Hi Craig & Mark,
>
> Right now it doesn't support streaming. I didn't want to bloat up the number of keywords & arguments to support streaming, and it would also make the documentation very confusing.
>
> What exactly do you want to do with streaming?

hi Chris,

In my case, I need to support reading and writing from/to an existing
file format which contains an embedded, zlib-compressed, data block.
These blocks can be very large - too large to hold in memory. Hence,
I need to be able to read it out in chunks, decompressing as I go.

best,
Mark

Subject: Re: How to Zip cross-platform from IDL?
Posted by Craig Markwardt on Thu, 18 Apr 2013 16:23:11 GMT
View Forum Message <> Reply to Message

On Thursday, April 18, 2013 9:40:08 AM UTC-4, Chris Torrence wrote:
> Hi Craig & Mark,
>
>
>
> Right now it doesn't support streaming. I didn't want to bloat up the number of keywords & arguments to support streaming, and it would also make the documentation very confusing.
>
>
>
> What exactly do you want to do with streaming? The problem with doing streamed compress/uncompress is that you need to chunk up the data in somewhat arbitrary ways. In particular, when doing uncompression, the uncompressor has to wait until it has gotten enough data to uncompress a particular chunk, while with compression, you might call it several times before it actually returns a chunk of compressed data. Very messy, especially for a vector-based language like IDL.
>
>
>
> Is it possible that you could simply chunk up your data yourself, say in 1KB or 1MB chunks, compress those using the new zlib_compress(), send them "across", and then uncompress them? Even though you're doing the chunking yourself, I would bet that it would be simpler to code that up than to use any sort of streaming functions that we might provide.

My situation is similar to Mark's. I have already-gzipped files that I want to read from a disk or over the network, without unzipping onto disk (they expand to multiple gigabytes). You might say, "who cares, disks are cheap," but disks are also slow, and I'm often interested in only the first few hundred kilobytes or less of the uncompressed data. I can use 'gzip' as a stream, but that only works under Unix-like operating systems.

I agree that the interface for streaming is more complicated to design than whole files, but I don't

think it's that complicated. Add an input hopper, an output hopper, and make it up to the client to feed the beast.

Craig
