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