
Subject: Re: How to Zip cross-platform from IDL?
Posted by [David Fanning](#) on Fri, 22 Feb 2013 13:44:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

Brian J. Daniel writes:

> 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?

I haven't solved this problem, either, with my own KML files. It would be wonderful to have the IDL zip API exposed so it can be done from within IDL. Please let us know if you come up with something.

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: How to Zip cross-platform from IDL?
Posted by [Klemen](#) on Fri, 22 Feb 2013 14:37:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

Perhaps SPAWN + COMPACT
<http://technet.microsoft.com/en-us/library/bb490884.aspx>

Subject: Re: How to Zip cross-platform from IDL?
Posted by [Heinz Stege](#) on Fri, 22 Feb 2013 17:11:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Brian.

On Fri, 22 Feb 2013 05:40:16 -0800 (PST), Brian J. Daniel wrote:

> 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

I think there should be a solution. First let me say, that the Windows version of 7-Zip comes together with a command line version: 7z.exe. The help file says, that it uses the same dll as the 7-Zip File Manager itself. So it should do the job.

Alternatively you can try the (good old) "GNU utilities for Win32". Download UnxUtils.zip from <http://sourceforge.net/projects/unxutils/files/>. There is gzip and gunzip support for Windows 32.

And last but not least there are the `idl_tools` from Randall Frank and Ronn Kling. They make system routines for gzip, gunzip, zip and more directly available to IDL by a `dln` file. I just tested the socket routines from this library. They work great. You can download the `dln` and `dll` files for windows from www.exelisvis.com. The authors say, that the routines run on both, Windows and Unix. However I can not say, where to find the sources for Unix. In Feb. 2010 they were available from <http://www.kilvarock.com/p1.htm>. This site however, today contains advertisements only. (Maybe, www.archive.org can help you.)

HTH, Heinz

Subject: Re: How to Zip cross-platform from IDL?

Posted by chris_torrence@NOSPAM on Fri, 22 Feb 2013 17:50:11 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi all,

We are planning on exposing ZIP functionality in an upcoming release. In the meantime, there is an undocumented way to create zipped files from IDL, using the KMZ functionality built into new graphics.

Try the following:

```
void = {IDLitWriteKML} ; just get the code compiled  
void = IDLKML_SaveKMZ(outputfile, inputfileArray)
```

For example:

```
void = IDLKML_SaveKMZ('myfile.kmz', $  
  ['myfile.kml','myimage1.png','myimage2.png'])
```

I would not recommend putting this code into a shipping product, as it is certainly not documented or supported. But regardless, it should work on all platforms.

Cheers,
Chris
ExelisVIS

Subject: Re: How to Zip cross-platform from IDL?
Posted by [Mark Piper](#) on Fri, 22 Feb 2013 18:09:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Friday, February 22, 2013 10:50:11 AM UTC-7, Chris Torrence wrote:

> We are planning on exposing ZIP functionality in an upcoming release.

To elaborate: we're looking at supporting zip, gzip and tar in IDL. I'm planning this for 8.2.3 (late spring), but it may slip to 8.3 (early fall).

Also note: I just tried Chris' example, and the input files are moved (not copied) into the zip file.

mp

Subject: Re: How to Zip cross-platform from IDL?
Posted by [David Fanning](#) on Fri, 22 Feb 2013 18:12:52 GMT
[View Forum Message](#) <> [Reply to Message](#)

Mark Piper writes:

> Also note: I just tried Chris' example, and the input files are moved (not copied) into the zip file.

Even better, from my point of view. :-)

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>

Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: How to Zip cross-platform from IDL?

Posted by [David Fanning](#) on Sat, 23 Feb 2013 15:44:49 GMT

[View Forum Message](#) <> [Reply to Message](#)

Chris Torrence writes:

> We are planning on exposing ZIP functionality in an upcoming release. In the meantime, there is an undocumented way to create zipped files from IDL, using the KMZ functionality built into new graphics.

>

> Try the following:

>

> void = {IDLitWriteKML} ; just get the code compiled

> void = IDLKML_SaveKMZ(outputfile, inputfileArray)

>

> For example:

> void = IDLKML_SaveKMZ('myfile.kmz', \$

> ['myfile.kml','myimage1.png','myimage2.png'])

>

> I would not recommend putting this code into a shipping product, as it is certainly not documented or supported. But regardless, it should work on all platforms.

I understand your recommendation, but this works so nicely I couldn't resist adding the functionality to my own programs that create KML files. In the process I discovered I was using absolute URLs to the image files instead of relative URLs. This has been fixed, and you can find all the updated files (e.g., cgImage2KML, cgCBar2KML, cgKML_File) in today's Coyote Library:

http://www.idlcoyote.com/programs/zip_files/coyoteprograms.zip

I've also updated the Google Earth Image code in the Coyote Plot Gallery to create a KMZ file:

http://www.idlcoyote.com/gallery/index.html#GE_IMAGE

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>

Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: How to Zip cross-platform from IDL?

Posted by [Brian Daniel](#) on Mon, 25 Feb 2013 14:52:24 GMT

[View Forum Message](#) <> [Reply to Message](#)

This is wonderful! Thank you so much, Chris. I'm happy that it will be officially a part of an upcoming release.

I'm troubleshooting the ability of utilizing subdirectories. For example, I have several different types of images that the KML is referencing, and even when I use relative paths, the KMZ process (from earlier in post) zips all images and places them in the same zipped directory. Its not that big of a deal, but when implementing this into a "feature", it should be kept in mind.

Thanks again!

(And thanks for your tools, David. I used your tool to check myself. When this method fails, it just saves a binary ntuser.dat file. But hey, can't complain for an undocumented feature!)

-B

On Saturday, February 23, 2013 10:44:49 AM UTC-5, David Fanning wrote:

> Chris Torrence writes:

>

>

>

>> We are planning on exposing ZIP functionality in an upcoming release. In the meantime, there is an undocumented way to create zipped files from IDL, using the KMZ functionality built into new graphics.

>

>>

>

>> Try the following:

>

>>

>

>> void = {IDLitWriteKML} ; just get the code compiled

>

```

>> void = IDLKML_SaveKMZ(outputfile, inputfileArray)
>
>>
>
>> For example:
>
>> void = IDLKML_SaveKMZ('myfile.kmz', $
>
>>   ['myfile.kml','myimage1.png','myimage2.png'])
>
>>
>
>> I would not recommend putting this code into a shipping product, as it is certainly not
documented or supported. But regardless, it should work on all platforms.
>
>
>
> I understand your recommendation, but this works so nicely I couldn't
>
> resist adding the functionality to my own programs that create KML
>
> files. In the process I discovered I was using absolute URLs to the
>
> image files instead of relative URLs. This has been fixed, and you can
>
> find all the updated files (e.g., cgImage2KML, cgCBar2KML, cgKML_File)
>
> in today's Coyote Library:
>
>
>
>   http://www.idlcoyote.com/programs/zip\_files/coyoteprograms.z ip
>
>
>
> I've also updated the Google Earth Image code in the Coyote Plot Gallery
>
> to create a KMZ file:
>
>
>
>   http://www.idlcoyote.com/gallery/index.html#GE\_IMAGE
>
>
>
> Cheers,
>
>
>

```

>
> David
>
>
>
> --
>
> David Fanning, Ph.D.
>
> Fanning Software Consulting, Inc.
>
> Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>
>
> Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: How to Zip cross-platform from IDL?
Posted by [timothyja123](#) on Thu, 28 Feb 2013 02:41:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Saturday, February 23, 2013 5:09:43 AM UTC+11, Mark Piper wrote:
> On Friday, February 22, 2013 10:50:11 AM UTC-7, Chris Torrence wrote:
>
>
>
>> We are planning on exposing ZIP functionality in an upcoming release.
>
>
>
> To elaborate: we're looking at supporting zip, gzip and tar in IDL. I'm planning this for 8.2.3 (late spring), but it may slip to 8.3 (early fall).
>
>
>
> Also note: I just tried Chris' example, and the input files are moved (not copied) into the zip file.
>
>
>
> mp

I'm really looking forward to this feature. Currently I must untar files so that I can reload there contents in IDL. The tars have around 2000 small files in them and I've just discovered that the Windows NTFS sucks at creating this many files in a single directory (takes around 12 second on my high end machine). On even a low end linux machine it is pretty much instant.

Anyway being able to load these files directly into memory would be a great improvement.

Subject: Re: How to Zip cross-platform from IDL?

Posted by chris_torrence@NOSPAM on Tue, 16 Apr 2013 04:50:04 GMT

[View Forum Message](#) <> [Reply to Message](#)

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
