Subject: reading gziped CDFs
Posted by erikb on Thu, 27 Feb 2003 04:50:21 GMT
View Forum Message <> Reply to Message

Hi all,

I've got a question that google can't answer, so I'm posing it to you:

I have gziped CDF files that I would like to open without having to manually unzip them for space considerations. I know that IDL can handle opening gziped files with the /compress keyword to the open routines, but that doesn't help with a CDF. I see that my ver. of IDL (5.4) has a CDF_compression command, but it seems that this is an internal compression method within the CDF structure. So... is there a way to get IDL to understand that I have a gzipped CDF? I'm thinking I need to kludge CDF_open somehow. I would only need to read these files, I would never write/append to them.

Thanks for any ideas
-Erik

p.s. I suppose an alternative is to write wrapper open/close functions that would SPAWN a gunzip to a temp file and then delete it on close. Anyone know a fast way of making unique temp filenames? (I'm on a linux system, btw.)

Subject: Re: reading gziped CDFs
Posted by Kenneth P. Bowman on Thu, 27 Feb 2003 12:22:04 GMT
View Forum Message <> Reply to Message

In article <b3k5k4cva1@news3.bu.edu>, erikb <erikb@bu.edu> wrote:

> p.s. I suppose an alternative is to write wrapper open/close functions
> that would SPAWN a gunzip to a temp file and then delete it on close.
> Anyone know a fast way of making unique temp filenames? (I'm on a linux
> system, btw.)

Try the mktemp command.

Ken Bowman

Subject: Re: reading gziped CDFs
Posted by Rick Towler on Thu, 27 Feb 2003 17:26:45 GMT
View Forum Message <> Reply to Message

"erikb" wrote ...

> I have gziped CDF files that I would like to open without having to
> manually unzip them for space considerations. I know that IDL can
> handle opening gziped files with the /compress keyword to the open
> routines, but that doesn't help with a CDF. I see that my ver. of IDL
> (5.4) has a CDF_compression command, but it seems that this is an
> internal compression method within the CDF structure.

> So... is there a way to get IDL to understand that I have
> a gzipped CDF?

No.

> I'm thinking I need to kludge CDF_open somehow.

Since (net)CDF functionality is provided via .dlms your ability to kludge anything will be rather limited unless you roll your own.

I think your only option is the unzip -> use -> zip approach you have already thought of. Ken suggested using mktemp() to create unique file names although that is a C library function and isn't native to IDL. An alternative would be to use the last 8 or so digits of systime(/seconds) to create your filename. This approach has limits (you only get 1 unique name per second) but I think it would be fine for what you want to do.

-Rick

Subject: Re: reading gziped CDFs
Posted by Mark Hadfield on Thu, 27 Feb 2003 18:40:25 GMT
View Forum Message <> Reply to Message

"erikb" <erikb@bu.edu> wrote in message news:b3k5k4cva1@news3.bu.edu...
> Hi all,
>
> I've got a question that google can't answer, so I'm posing it to
> you:
>
> I have gziped CDF files that I would like to open without having to
> manually unzip them for space considerations. I know that IDL can
> handle opening gziped files with the /compress keyword to the open
> routines, but that doesn't help with a CDF. I see that my ver. of
> IDL (5.4) has a CDF_compression command, but it seems that this is
> an internal compression method within the CDF structure.

Yes. The CDF format supports compression of each variable internally. I haven't used this feature for a long time, but I see

from the IDL documentation that it allows a number of compression methods including GZIP, so it should be pretty effective. Is there any possibility of (re)writing your files using CDF compression? Then you might not need to gzip them.

--
Mark Hadfield "Ka puwaha te tai nei, Hoea tatou"
m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)

Subject: Re: reading gziped CDFs
Posted by K. Bowman on Thu, 27 Feb 2003 18:43:09 GMT
View Forum Message <> Reply to Message

In article <b3lhp2$1j8e$1@nntp6.u.washington.edu>,
"Rick Towler" <rtowler@u.washington.edu> wrote:

> Ken suggested using mktemp() to create unique file
> names although that is a C library function and isn't native to IDL.

Actually, on my BSD-based system (OS X) mktemp has a command line interface. Since he was thinking of spawning a gunzip command, he could easily spawn a mktemp command first to generate a filename.

It would be safer, but not as portable as using the system clock.

Regards, Ken

bowman> mktemp /tmp/idl.XXXXX
/tmp/idl.9OIp9
bowman> ll /tmp
total 0
-rw------- 1 bowman wheel 0 Feb 27 12:36 idl.9OIp9

MKTEMP(1) System General Commands Manual
MKTEMP(1)

NAME
mktemp - make temporary file name (unique)

SYNOPSIS
mktemp [-d] [-q] [-t prefix] [-u] template ...
mktemp [-d] [-q] [-u] -t prefix

DESCRIPTION

The mktemp utility takes each of the given file name templates and over-writes a portion of it to create a file name. This file name is unique and suitable for use by the application. The template may be any file name with some number of `Xs' appended to it, for example /tmp/temp.XXXX. The trailing `Xs' are replaced with the current process number and/or a unique letter combination. The number of unique file names mktemp can return depends on the number of `Xs' provided; six `Xs' will result in mktemp testing roughly 26 ** 6 combinations.

If mktemp can successfully generate a unique file name, the file is created with mode 0600 (unless the -u flag is given) and the filename is printed to standard output.