Subject: Crowd support wanted for adding metadata to READ_PNG and WRITE_PNG
Posted by andrewcool777 on Mon, 05 Sep 2016 03:11:11 GMT
View Forum Message <> Reply to Message

Hi All,

The PNG image format is a wonderous thing. Lossless, but with user selectable variable compression, 8 or 16bit image data, and most importantly, the ability to
have user-defined metadata!

Yet for some reason, in their infinite wisdom, RSI never provided the functionality to read or write the metadata when they wrote the routines! :-(

I run a number of airglow cameras, provided by the great guys at Keo Scientific,
and the images are saved as PNG files, with a basic set of metadata. Now I've been able to write IDL code to extract that metadata, and write PNG files with (extended, modified) metadata, as reported previously, but it's quite (read _very_) slow, and would definitely be better off done within the IDL Read_PNG and Write_PNG routines!

However, as I'm the only IDL user on the planet to request this feature, I'm told by Harris Support that it won't happen unless more users put their hands up and
say "Me too, please!"

So here's my plea to you. If you think it might be a Good Thing to have IDL's
own PNG routines handle the reading and writing of user supplied metadata, please submit a Feature Request to support@harris.com to this effect.

Many Hosannas will be said in your name for doing so...

Regards,

Andrew Cool
www.skippysky.com.au (for all your astronomy weather forecast needs...)

Subject: Re: Crowd support wanted for adding metadata to READ_PNG and WRITE_PNG
Posted by markb77 on Tue, 06 Sep 2016 11:26:47 GMT
View Forum Message <> Reply to Message

On Monday, September 5, 2016 at 5:11:17 AM UTC+2, andrew...@gmail.com wrote:
> Hi All,
>
> The PNG image format is a wonderous thing. Lossless, but with user selectable variable compression, 8 or 16bit image data, and most importantly, the ability to
> have user-defined metadata!

>
> Yet for some reason, in their infinite wisdom, RSI never provided the functionality to read or write the metadata when they wrote the routines! :-(
>
> I run a number of airglow cameras, provided by the great guys at Keo Scientific,
> and the images are saved as PNG files, with a basic set of metadata. Now I've been able to write IDL code to extract that metadata, and write PNG files with (extended, modified) metadata, as reported previously, but it's quite (read _very_) slow, and would definitely be better off done within the IDL Read_PNG and Write_PNG routines!
>
> However, as I'm the only IDL user on the planet to request this feature, I'm told by Harris Support that it won't happen unless more users put their hands up and
> say "Me too, please!"
>
> So here's my plea to you. If you think it might be a Good Thing to have IDL's
> own PNG routines handle the reading and writing of user supplied metadata, please
> submit a Feature Request to support@harris.com to this effect.
>
> Many Hosannas will be said in your name for doing so...
>
>
> Regards,
>
> Andrew Cool
> www.skippysky.com.au (for all your astronomy weather forecast needs...)

This is not exactly what you're looking for, but just out of curiosity.. Have you considered using the equivalent routines from Python? Are these not, essentially, available within IDL as of the latest releases?

If you are distributing an IDL application, perhaps this wouldn't work for you? Or, would it? What are the implications of using the IDL-Python bridge for distributed IDL applications which are meant to be run via the virtual machine?

best
Mark

Subject: Re: Crowd support wanted for adding metadata to READ_PNG and WRITE_PNG
Posted by Markus Schmassmann on Tue, 06 Sep 2016 11:48:04 GMT
View Forum Message <> Reply to Message

On 09/05/2016 05:11 AM, andrewcool777@gmail.com wrote:
> The PNG image format is a wonderous thing. Lossless, but with user
> selectable variable compression, 8 or 16bit image data, and most
> importantly, the ability to have user-defined metadata!
>

> Yet for some reason, in their infinite wisdom, RSI never provided the
> functionality to read or write the metadata when they wrote the
> routines! :-(
>
> I run a number of airglow cameras, provided by the great guys at Keo
> Scientific, and the images are saved as PNG files, with a basic set
> of metadata. Now I've been able to write IDL code to extract that
> metadata, and write PNG files with (extended, modified) metadata, as
> reported previously, but it's quite (read _very_) slow, and would
> definitely be better off done within the IDL Read_PNG and Write_PNG
> routines!
>
> However, as I'm the only IDL user on the planet to request this
> feature, I'm told by Harris Support that it won't happen unless more
> users put their hands up and say "Me too, please!"
>
> So here's my plea to you. If you think it might be a Good Thing to
> have IDL's own PNG routines handle the reading and writing of user
> supplied metadata, please submit a Feature Request to
> support@harris.com to this effect.
>
> Many Hosannas will be said in your name for doing so...

Hi Andrew

Having experienced tediously slow read in of FITS-files/headers and having spent some time optimizing these read-in routines i wonder, whether you could increase your PNG read-in routine speed by up to 2 order of magnitudes by providing the structure templates for the metadata instead of using many create_struct() calls on runtime. But obviously i can't tell without seeing the code.

Markus

Subject: Re: Crowd support wanted for adding metadata to READ_PNG and WRITE_PNG
Posted by Dick Jackson on Tue, 06 Sep 2016 15:31:25 GMT
View Forum Message <> Reply to Message

On Tuesday, 6 September 2016 04:26:50 UTC-7, superchromix wrote:
> On Monday, September 5, 2016 at 5:11:17 AM UTC+2, andrew...@gmail.com wrote:
>> Hi All,
>>
>> The PNG image format is a wonderous thing. Lossless, but with user selectable variable compression, 8 or 16bit image data, and most importantly, the ability to
>> have user-defined metadata!
>>
>> Yet for some reason, in their infinite wisdom, RSI never provided the functionality to read or

write the metadata when they wrote the routines! :-(
>>
>> I run a number of airglow cameras, provided by the great guys at Keo Scientific,
>> and the images are saved as PNG files, with a basic set of metadata. Now I've been able to write IDL code to extract that metadata, and write PNG files with (extended, modified) metadata, as reported previously, but it's quite (read _very_) slow, and would definitely be better off done within the IDL Read_PNG and Write_PNG routines!
>>
>> However, as I'm the only IDL user on the planet to request this feature, I'm told by Harris Support that it won't happen unless more users put their hands up and
>> say "Me too, please!"
>>
>> So here's my plea to you. If you think it might be a Good Thing to have IDL's
>> own PNG routines handle the reading and writing of user supplied metadata, please
>> submit a Feature Request to support@harris.com to this effect.
>>
>> Many Hosannas will be said in your name for doing so...
>>
>>
>> Regards,
>>
>> Andrew Cool
>> www.skippysky.com.au (for all your astronomy weather forecast needs...)
>
> This is not exactly what you're looking for, but just out of curiosity.. Have you considered using the equivalent routines from Python? Are these not, essentially, available within IDL as of the latest releases?
>
> If you are distributing an IDL application, perhaps this wouldn't work for you? Or, would it? What are the implications of using the IDL-Python bridge for distributed IDL applications which are meant to be run via the virtual machine?
>
> best
> Mark

I can confirm that, at least with IDL 8.5.1 on Windows 10, IDL VM calls to Python work just fine! This discovery made me very happy.

Cheers,
-Dick

Dick Jackson Software Consulting Inc.
Victoria, BC, Canada --- http://www.d-jackson.com

Subject: Re: Crowd support wanted for adding metadata to READ_PNG and WRITE_PNG

On Tuesday, 6 September 2016 20:56:50 UTC+9:30, superchromix wrote:
> On Monday, September 5, 2016 at 5:11:17 AM UTC+2, andrew...@gmail.com wrote:
>> Hi All,
>>
>> The PNG image format is a wonderous thing. Lossless, but with user selectable variable compression, 8 or 16bit image data, and most importantly, the ability to
>> have user-defined metadata!
>>
>> Yet for some reason, in their infinite wisdom, RSI never provided the functionality to read or write the metadata when they wrote the routines! :-(
>>
>> I run a number of airglow cameras, provided by the great guys at Keo Scientific,
>> and the images are saved as PNG files, with a basic set of metadata. Now I've been able to write IDL code to extract that metadata, and write PNG files with (extended, modified) metadata, as reported previously, but it's quite (read _very_) slow, and would definitely be better off done within the IDL Read_PNG and Write_PNG routines!
>>
>> However, as I'm the only IDL user on the planet to request this feature, I'm told by Harris Support that it won't happen unless more users put their hands up and
>> say "Me too, please!"
>>
>> So here's my plea to you. If you think it might be a Good Thing to have IDL's
>> own PNG routines handle the reading and writing of user supplied metadata, please
>> submit a Feature Request to support@harris.com to this effect.
>>
>> Many Hosannas will be said in your name for doing so...
>>
>>
>> Regards,
>>
>> Andrew Cool
>> www.skippysky.com.au (for all your astronomy weather forecast needs...)
>
> This is not exactly what you're looking for, but just out of curiosity.. Have you considered using the equivalent routines from Python? Are these not, essentially, available within IDL as of the latest releases?
>
> If you are distributing an IDL application, perhaps this wouldn't work for you? Or, would it? What are the implications of using the IDL-Python bridge for distributed IDL applications which are meant to be run via the virtual machine?
>
> best
> Mark

Python. Exactly what Harris said. So why then am I paying bucketloads of money for IDL? I'd like IDL to do the job, not get fobbed off onto some 3rd party, flakey software that has it's own set of

bugs and quirks to contend with.

Whilst I'm not distributing the code externally, neither do I want to force all my colleagues to install Python - a task made even more difficult by the Defence
environment that we work in.

I'm of the opinion that IDL, as a very expensive, graphics orientated package
should be able to correctly handle a very common, standard image format, without passing the buck.

If I have to start using Python, perhaps I don't need IDL, and Harris are shooting themselves in the foot?

Andrew

Subject: Re: Crowd support wanted for adding metadata to READ_PNG and WRITE_PNG
Posted by andrewcool777 on Tue, 27 Sep 2016 00:33:56 GMT
View Forum Message <> Reply to Message

On Tuesday, 6 September 2016 21:18:06 UTC+9:30, Markus Schmassmann wrote:
> On 09/05/2016 05:11 AM, andrewcool777@gmail.com wrote:
>> The PNG image format is a wonderous thing. Lossless, but with user
>> selectable variable compression, 8 or 16bit image data, and most
>> importantly, the ability to have user-defined metadata!
>>
>> Yet for some reason, in their infinite wisdom, RSI never provided the
>> functionality to read or write the metadata when they wrote the
>> routines! :-(
>>
>> I run a number of airglow cameras, provided by the great guys at Keo
>> Scientific, and the images are saved as PNG files, with a basic set
>> of metadata. Now I've been able to write IDL code to extract that
>> metadata, and write PNG files with (extended, modified) metadata, as
>> reported previously, but it's quite (read _very_) slow, and would
>> definitely be better off done within the IDL Read_PNG and Write_PNG
>> routines!
>>
>> However, as I'm the only IDL user on the planet to request this
>> feature, I'm told by Harris Support that it won't happen unless more
>> users put their hands up and say "Me too, please!"
>>
>> So here's my plea to you. If you think it might be a Good Thing to
>> have IDL's own PNG routines handle the reading and writing of user
>> supplied metadata, please submit a Feature Request to
>> support@harris.com to this effect.
>>

>> Many Hosannas will be said in your name for doing so...
> Hi Andrew
>
> Having experienced tediously slow read in of FITS-files/headers and
> having spent some time optimizing these read-in routines i wonder,
> whether you could increase your PNG read-in routine speed by up to 2
> order of magnitudes by providing the structure templates for the
> metadata instead of using many create_struct() calls on runtime. But
> obviously i can't tell without seeing the code.
>
> Markus

Hi Markus,

PNG Metadata is nothing like FITS. The metadata has to be read in chunks, byte by byte, and the image data has to be compressed/decompressed byte by byte on the fly, which is the big consumer of time. There are no create_struct() calls involved.

Andrew

Subject: Re: Crowd support wanted for adding metadata to READ_PNG and WRITE_PNG
Posted by penteado on Thu, 08 Dec 2016 00:54:59 GMT
View Forum Message <> Reply to Message

On Sunday, September 4, 2016 at 8:11:17 PM UTC-7, andrew...@gmail.com wrote:
> So here's my plea to you. If you think it might be a Good Thing to have IDL's
> own PNG routines handle the reading and writing of user supplied metadata, please
> submit a Feature Request to support@harris.com to this effect.
>
> Many Hosannas will be said in your name for doing so...

A bit of a late reply, but I can add to this request. I work with PNGs often, and metadata support would definitely be useful to me in some applications now and in the future.

As to the Python bridge solution, I tend to agree with Andrew: While the bridge is very useful and opens up many new possibilities, I see it as mainly a way to interact with complex, problem-specific preexisting Python libraries. It should not be relied upon for basic tasks that should be handled in IDL. In most applications I make, the code is to be distributed to other users, I cannot ask them to have a working Python and IDL-Python bridge (tricky to set up and dependent on having a recent IDL version and license) just because there is a minor task in the code that is being handed off to Python. Which, besides the added difficulty for the users, reinforces the ever more common question these days: "why did you use IDL instead of Python?". One big reason I have is avoiding the hassle of setting up a working Python environment with all the needed libraries and no conflicts with the other Python environments used in other applications. We do not want Python to be a dependency for IDL applications.
