
Subject: How to create a sav file for a system based on catalyst

Posted by [Jie Zhou](#) on Mon, 03 Mar 2014 09:46:24 GMT

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

I coded a system based on catalyst. And many class given in catalyst lib has been used. Now I try to create a sav file for my system. It seems that I should resolve all class defined in the catalyst and used in my system. Is there some other simple method to finish work?

Cheers,

Jie

Subject: Re: How to create a sav file for a system based on catalyst

Posted by [David Fanning](#) on Mon, 03 Mar 2014 13:14:28 GMT

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

Jie Zhou writes:

> I coded a system based on catalyst. And many class given in catalyst lib has been used. Now I try to create a sav file for my system. It seems that I should resolve all class defined in the catalyst and used in my system. Is there some other simple method to finish work?

Yes, objects are always a problem. I always say that `Resolve_All` resolves "almost all" routines, but it ignores objects. (I know about the `CLASS` keyword, but in practice that seldom helps me.)

What I do when I build a Catalyst application is to have Catalyst in one project and my application in other. Then I "build" both applications, use `Resolve_All` after doing so, and hope for the best. Sometimes that works well. :-)

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 create a sav file for a system based on catalyst

Posted by [Phillip Bitzer](#) on Mon, 03 Mar 2014 16:21:17 GMT

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

On Monday, March 3, 2014 7:14:28 AM UTC-6, David Fanning wrote:

>

> Yes, objects are always a problem. I always say that Resolve_All
>
> resolves "almost all" routines, but it ignores objects. (I know about
>
> the CLASS keyword, but in practice that seldom helps me.)
>

Hmm. I find quite the opposite - class _does_ help me - although I'm not maintaining projects quite as extensive as yours.

To build my applications, I have a "make" file that I run (as a script) in a fresh session of IDL.

All (actually, most) of the routines are in one folder, so the make file starts with this:

```
files = FILE_SEARCH('path_to_dir/*.pro')
FOR i=0, N_ELEMENTS(files)-1 DO RESOLVE_ROUTINE, FILE_BASENAME(files[i], '.pro'),
/EITHER
```

There are some routines not picked up by this, including my generic object handlers (s/o to Mike Galloy), so I also have things like:

```
RESOLVE_ROUTINE, 'pmb_object_event_handler'
RESOLVE_ROUTINE, 'pmb_object_cleanup'
```

Finally, like David said, objects (and structure definitions) are not picked up by RESOLVE_ROUTINE, so I have to include things like this too:

```
RESOLVE_ALL, CLASS='cgCmdWindow', /CONTINUE
```

Finally, I have the line:

```
SAVE, /ROUTINES, filename=save_filename
```

It's not quite as easy as the "Build Project," and it takes a little hand holding (when I add a new object, I have to remember to put it in the make file), but I find it gives me more flexibility and, more importantly, it works.

Subject: Re: How to create a sav file for a system based on catalyst

Posted by [Jie Zhou](#) on Fri, 07 Mar 2014 08:46:24 GMT

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

On Monday, March 3, 2014 5:21:17 PM UTC+1, Phillip Bitzer wrote:

> On Monday, March 3, 2014 7:14:28 AM UTC-6, David Fanning wrote:
>
>
>
>>
>
>> Yes, objects are always a problem. I always say that Resolve_All
>

```

>>
>
>> resolves "almost all" routines, but it ignores objects. (I know about
>
>>
>
>> the CLASS keyword, but in practice that seldom helps me.)
>
>>
>
>
> Hmm. I find quite the opposite - class _does_ help me - although I'm not maintaing projects
quite as extensive as yours.
>
>
>
> To build my applications, I have a "make" file that I run (as a script) in a fresh session of IDL.
>
>
>
> All (actually, most) of the routines are in one folder, so the make file starts with this:
>
>
>
> files = FILE_SEARCH('path_to_dir/*.pro')
>
> FOR i=0, N_ELEMENTS(files)-1 DO RESOLVE_ROUTINE, FILE_BASENAME(files[i], '.pro'),
/EITHER
>
>
>
> There are some routines not picked up by this, including my generic object handlers (s/o to
Mike Galloy), so I also have things like:
>
> RESOLVE_ROUTINE, 'pmb_object_event_handler'
>
> RESOLVE_ROUTINE, 'pmb_object_cleanup'
>
>
>
> Finally, like David said, objects (and structure definitions) are not picked up by
RESOLVE_ROUTINE, so I have to include things like this too:
>
> RESOLVE_ALL, CLASS='cgCmdWindow', /CONTINUE
>
>
>

```

> Finally, I have the line:
>
> SAVE, /ROUTINES, filename=save_filename
>
>
>
> It's not quite as easy as the "Build Project," and it takes a little hand holding (when I add a new object, I have to remember to put it in the make file), but I find it gives me more flexibility and, more importantly, it works.

That's similar as my solution!
thanks.
