Subject: Re: structure arrays with different structures
Posted by Craig Markwardt on Sun, 09 Sep 2012 04:18:49 GMT
View Forum Message <> Reply to Message

On Saturday, September 8, 2012 9:08:39 PM UTC-4, Søren Frimann wrote:
> Hi all,
>
>
>
> I'm having some data that's gathered together in different structures and arrays, and I want to collect everything in a structure array in this manner:
>
>
>
> struct_template = { name:'', select:{} , obs:dblarr(2,100)}
>
>
>
> I want to put my data into struct_template, with name being a string, select being another structure, and obs being an array.
>
>
>
> I then want to make an array of structures using struct_template, but naturally with different sets of data.
>
>
>
> The issue is that the select structure does not always have the same skeleton, nor does the obs array always have the same size.
>
>
>
> The question basically is if there's any way of making an array of structures with the structures sharing the same keys, but with the values having different 'skeletons'

You want to use pointers. It's what pointers were invented for.

However, don't go overboard with pointers. It can create a managerial nightmare. I would advise you to keep the basic structure the same, so you can have an array of identical structures. Use pointers for your OBS field and other variations from entry to entry.

Craig

Subject: Re: structure arrays with different structures
Posted by Søren Frimann on Sun, 09 Sep 2012 11:33:32 GMT

Den søndag den 9. september 2012 06.18.49 UTC+2 skrev Craig Markwardt:
> On Saturday, September 8, 2012 9:08:39 PM UTC-4, Søren Frimann wrote:
>
>> Hi all,
>
>>
>
>>
>
>>
>
>> I'm having some data that's gathered together in different structures and arrays, and I want to collect everything in a structure array in this manner:
>
>>
>
>>
>
>>
>
>> struct_template = { name:'', select:{} , obs:dblarr(2,100)}
>
>>
>
>>
>
>>
>
>> I want to put my data into struct_template, with name being a string, select being another structure, and obs being an array.
>
>>
>
>>
>
>>
>
>> I then want to make an array of structures using struct_template, but naturally with different sets of data.
>
>>
>
>>
>
>>
>
>> The issue is that the select structure does not always have the same skeleton, nor does the

obs array always have the same size.
>
>>
>
>>
>
>>
>
>> The question basically is if there's any way of making an array of structures with the structures sharing the same keys, but with the values having different 'skeletons'
>
>
>
> You want to use pointers. It's what pointers were invented for.
>
>
>
> However, don't go overboard with pointers. It can create a managerial nightmare. I would advise you to keep the basic structure the same, so you can have an array of identical structures. Use pointers for your OBS field and other variations from entry to entry.
>
>
>
> Craig

Thanks for the pointer (very bad pun intended) :-)

-Søren

Subject: Re: structure arrays with different structures
Posted by Paul Van Delst[1] on Mon, 10 Sep 2012 13:43:54 GMT
View Forum Message <> Reply to Message

Why not use a list of structures? (Assuming you have IDL v8.x).

Just add each structure to the list as it's created.

If you can avoid using pointers (directly at least), you should (IMO).

cheers,

paulv

On 09/08/12 21:08, Sï¿½ren Frimann wrote:
> Hi all,
>

> I'm having some data that's gathered together in different structures
> and arrays, and I want to collect everything in a structure array in
> this manner:
>
> struct_template = { name:'', select:{} , obs:dblarr(2,100)}
>
> I want to put my data into struct_template, with name being a string,
> select being another structure, and obs being an array.
>
> I then want to make an array of structures using struct_template, but
> naturally with different sets of data.
>
> The issue is that the select structure does not always have the same
> skeleton, nor does the obs array always have the same size.
>
> The question basically is if there's any way of making an array of
> structures with the structures sharing the same keys, but with the
> values having different 'skeletons'
>
> I hope I'm making sense
>
> Cheers, Sï¿½ren
