
Subject: Re: Inside rectangle

Posted by [Rick Towler](#) on Tue, 02 Jul 2002 21:49:11 GMT

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

"Neil Talsania" <neil.talsania@kodak.com> wrote > Now I think I want to make a structure to hold the 4 corner coordinates.

> Then I need to create an array of these structures, but I want that array to

> be able to grow. Then I will loop through that array (For loop) calling a

> function that will check to see if the point is inside the coordinates.

>

> So the crux of my problem (I think) is that I need the array size to

> increase every time the user adds a set. But I dont know ho to do this.

Reimar's suggestion will work but if you are carting this data around your application in a structure, at some point you'll want to use a pointer to reference your array of structures.

But better yet, why not use a linked list? Then you can easily add and remove individual rectangles and not have to deal with the details. You can download David Fanning's `linkedlist__define.pro` from his website (www.dfanning.com). Since the linked list is an object it is simple to cart around in a structure. His object returns a pointer to the data contained in a node and you will need to be aware of precedence when dereferencing the pointer:

imagine 'll' is my linked list object with all of my rectangle structures I can get a pointer 'p' to the first structure in the list:

```
p = ll->get_item(0)
```

To get the data that the pointer refers to I need to use parentheses to ensure that IDL dereferences the pointer p before trying to access the structure elements:

```
IDL> print,(*p).coord  
0.000000  0.000000  0.000000  0.000000
```

-Rick
