

---

Subject: Re: losing pointers when concatenating array of structures

Posted by [Heinz Stege](#) on Mon, 09 Jan 2017 10:10:27 GMT

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

---

Well, pointers are tricky sometimes. I will try to explain. The statement

```
structarray=struct
```

in the first loop-path creates a new structure with a new pointer variable pointing to the `_same_` heap variable as `struct.array`. The heap variable is `_not_` copied.

Undefine is not an IDL routine. I assume you have downloaded this procedure from the Coyote library. In the second loop path `undefine` destroys the `struct.array` pointer. I.e. it deletes the heap variable pointed to by `struct.array` as well as `structarray.array`.

Your fix with `temporary(struct)` moves the structure from `struct` to `structarray`. The `undefine` procedure can't delete the heap variable anymore, because `struct` is undefined at that time.

I'm not sure, that I use the right terms in my explanation. I hope, it gets clear for you yet.

Heinz

---