

## Re: Template collection classes

---

*Source:* <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.mfc/2007-08/msg01730.html>

---

- *From:* "David Webber" <dave@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
  - *Date:* Mon, 27 Aug 2007 22:58:16 +0100
- 

"marcomb" <marcomb@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message  
<news:9772B215-BED7-4E85-B55B-8BC712C9D180@xxxxxxxxxxxxxxxxxxxxxxxx>

```
CArray<CPoint, CPoint&> PointArray;
```

This hold a collection of CPoint's. Not references to CPoint's or pointers to CPoint's.

```
PointArray.Add(aPoint);
```

This makes a copy in the array.

```
so we pass aPoint by reference, but still we create a copy of it in  
PointArray
```

Yes.

```
CArray<CPoint, CPoint> PointArray;
```

I don't think there is any need to do that. The second template argument is to do with how things are referenced; the first argument is the thing stored.

```
CPoint aPoint = new CPoint(...paramlist...);  
...
```

That should not compile. (new returns a pointer).

Re: Template collection classes

```
CArray<CPoint, CPoint*> PointArray;
```

I would guess that will probably not work. use

```
CArray< CPoint, CPoint &>
```

```
CPoint aPoint = new CPoint(...paramlist...);
```

Again new returns a pointer.

```
PointArray.Add(aPoint);
```

what's the difference from

```
CTypePtrArray<CObList,CPoint*> PointArray;
```

Do you mean CTypedPtrArray ? I am not sure it is legal with a CObList, but CTypePtrArray is used typically with an array of pointers.

```
CPoint aPoint = new CPoint(...paramlist...);
```

Again new returns a pointer.

In practice I'd use `std::vector` instead of `CArray` these days. You can store an array of "things" (which need a copy constructor when you add them to the array) or an array of "pointers to things". It's your choice. In either case be careful about scope, and to delete things created with `new` at the appropriate point in order to avoid memory leaks.

Dave

--

David Webber

Author of 'Mozart the Music Processor'

<http://www.mozart.co.uk>

For discussion/support see

<http://www.mozart.co.uk/mzusers/maillinglist.htm>

.