

# How does STL map deal with scoping and memory persistence

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.vc/2005-07/msg00429.html>

---

- *From:* "Tommo" <[mark.thomson@xxxxxxxxxxxxxx](mailto:mark.thomson@xxxxxxxxxxxxxx)>
  - *Date:* 18 Jul 2005 21:47:04 -0700
- 

I am storing some key value pairs in a map as follows:

```
<code>
bool StaticDataCache::putMarket(uint8_t key,item_t* value)
{
typedef pair<uint8_t,item_t*> entry;
typedef map<uint8_t,item_t*>::iterator iter;

entry newEntry(key,value);
cout << "About to insert with key:" << key << endl;
pair<iter,bool> insertSuccess = marketsMap.insert(newEntry);

return true;
}
</code>
```

Now suprisingly to me this actually works when i was expecting it not to. As you may notice the pair , newEntry, is local to this function, which means, as i understand it, that once this function exits, the memory used by newEntry is reclaimed and hence the pair inserted into the map would be invalidated. Can anyone tell me why this is working. Does the STL pair handle a 'new' behind the scenes??

.

---

- *Follow-Ups:*
  - ◆ **[Re: How does STL map deal with scoping and memory persistence](#)**  
    ◇ *From:* Hendrik Schober
  - ◆ **[Re: How does STL map deal with scoping and memory persistence](#)**  
    ◇ *From:* Carl Daniel [VC++ MVP]
- Prev by Date: **[Re: help with "rolling my own" Timer class](#)**
- Next by Date: **[Drawing an Image over Another](#)**
- Previous by thread: **[help with "rolling my own" Timer class](#)**

How does STL map deal with scoping and memory persistence

- Next by thread: ***Re: How does STL map deal with scoping and memory persistence***
- Index(es):
  - ◆ *Date*
  - ◆ *Thread*