
Subject: A probably simple question about pick semantics

Posted by [peek](#) on Mon, 12 Dec 2011 15:58:41 GMT

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

I wanted to pass a Vector by reference to a template function, but the Vector is lost :

.cpp file

```
Vector<double> v;  
v << 1;
```

```
Test(v);
```

.h file

```
template <class T>  
void Test(T v)  
{  
    // Nothing done  
}
```

After calling to Test(), Vector v contents disappear.

When debugging you can see than before jumping to Test(), the compiler call to Vector pick constructor:

```
Vector(pick_ Vector& v)    { Pick(v); }
```

and before leaving Test(), the compiler calls Vector destructor:

```
~Vector() { ....
```

However if Test is called with a pointer to Vector, v contents remains.

```
Test(&v)
```

What could we do to call the templated function without losing Vector contents?

Thank you

Subject: Re: A probably simple question about pick semantics

Posted by [cbpporter](#) on Mon, 12 Dec 2011 16:08:41 GMT

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

Use a reference parameter: Test(T& v) or Test(const T& v).

Pick semantics kick in on a copy operation. Even without pick semantics using reference would be

advised, since otherwise you would be needlessly copying around data when passed to the function.

Subject: Re: A probably simple question about pick semantics

Posted by [peek](#) on Fri, 16 Dec 2011 21:35:21 GMT

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

Hello cbpporter

I have tried it doing it by reference but the problem remains .

```
template <class T>
```

```
void Test(T &v)
```

```
{
```

```
    // Nothing done
```

```
}
```

In fact I forgot to put the "&" in the sample code.

Is there a way to avoid this copy in a Vector passed by reference to a templated function?

Subject: Re: A probably simple question about pick semantics

Posted by [mirek](#) on Fri, 16 Dec 2011 21:49:38 GMT

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

peek wrote on Fri, 16 December 2011 16:35Hello cbpporter

I have tried it doing it by reference but the problem remains .

```
template <class T>
```

```
void Test(T &v)
```

```
{
```

```
    // Nothing done
```

```
}
```

In fact I forgot to put the "&" in the sample code.

Is there a way to avoid this copy in a Vector passed by reference to a templated function?

More complex (read "compilable") example would be needed...

In general, this has to work, because such templates are widespread in the U++ code (read "theide would not run")

Mirek

Subject: Re: A probably simple question about pick semantics

Posted by [peek](#) on Fri, 16 Dec 2011 23:26:48 GMT

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

Sorry... there is no problem.

Thank you all !!
