Subject: Moveable classes with virtual functions/Documentation error
Posted by hans on Sun, 16 Apr 2006 21:05:09 GMT
View Forum Message <> Reply to Message

Hi,
documentation states in srcdoc/Core/Moveable:

All of the following requirements should be fullfilled for moveable types:
- It does not have any virtual method nor virtual base class.

This is imho incorrect, as U++ itself uses moveable classes with virtual methods, e.g. class Value.

Don't know what's the case with virtual base classes

Subject: Re: Moveable classes with virtual functions/Documentation error
Posted by mirek on Mon, 17 Apr 2006 17:03:33 GMT
View Forum Message <> Reply to Message

hans wrote on Sun, 16 April 2006 17:05Hi,
documentation states in srcdoc/Core/Moveable:

All of the following requirements should be fullfilled for moveable types:
- It does not have any virtual method nor virtual base class.

This is imho incorrect, as U++ itself uses moveable classes with virtual methods, e.g. class Value.

Don't know what's the case with virtual base classes

First of all, no, Value does not have virtual methods (it stores a pointer to heap object with virtual methods, but that is something else).

Second, technically, it is true that there are no technical reasons for "normal" C++ implementation why polymorphic classes should not be moveable (while virtual base classes are not quite often AFAIK).

However, whole Moveable concept is walking on the thin ice. In practice, you really need just concrete types to be moveable; therefore virtual methods are not required. By excluding them, we might be makeing that ice little bit less thin, makeing "moveable" types closer to PODs, with no real disadvantage.

Mirek

Subject: Re: Moveable classes with virtual functions/Documentation error
Posted by hans on Mon, 17 Apr 2006 18:26:33 GMT

View Forum Message <> Reply to Message

Yes, you are right. Virtual methods are in embedded class Void inside Value, so no problem for Value.

But imho the restriction in practice is too restrictive, as all implementations I know off are storing a vptr (ptr to virtual methods table), so moveable should not be a problem.

I agree that in practice most of the time only concrecte
types need to be moveable, so playing safe is probably better.