

---

Subject: Re: Ptr improve

Posted by [kohait00](#) on Wed, 18 May 2011 09:40:17 GMT

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

---

for this matter

[http://www.cplusplus.com/reference/std/memory/auto\\_ptr/](http://www.cplusplus.com/reference/std/memory/auto_ptr/)

[http://www.boost.org/doc/libs/1\\_46\\_1/libs/smart\\_ptr/smart\\_ptr\\_r.htm](http://www.boost.org/doc/libs/1_46_1/libs/smart_ptr/smart_ptr_r.htm)

std::auto\_ptr -> UPP::One<>, sole ownership, not shared amongst others, pick semantic

Quote:

Conceptually, smart pointers are seen as owning the object pointed to, and thus responsible for deletion of the object when it is no longer needed.

The smart pointer library provides six smart pointer class templates:

scoped\_ptr <boost/scoped\_ptr.hpp> Simple sole ownership of single objects. Noncopyable.

scoped\_array <boost/scoped\_array.hpp> Simple sole ownership of arrays. Noncopyable.

shared\_ptr <boost/shared\_ptr.hpp> Object ownership shared among multiple pointers.

shared\_array <boost/shared\_array.hpp> Array ownership shared among multiple pointers.

weak\_ptr <boost/weak\_ptr.hpp> Non-owning observers of an object owned by shared\_ptr.

intrusive\_ptr <boost/intrusive\_ptr.hpp> Shared ownership of objects with an embedded reference count.

These templates are designed to complement the std::auto\_ptr template.

scoped\_ptr is a restricted version of One<>

shared\_ptr shares ownership of same object among multiple shared\_ptr instances (aka ref count, or Value). for this we dont have a 'clean' leightweight implementations, sth like Shared<> would be great.

weak\_ptr is a weak ref, pointing same stuff shared\_ptr already points to, comparable to Ptr<>, it doesnt hold ownership, just as Ptr<> doesnt, so we actually \*do\* have almost all of it.

maybe we really should consider to implement such a shared ownership container, sth like Shared<>