

NAME

update – mark object dirty

SYNOPSIS

```
#include <ShoreApp.h>
T *Ref<T>::update() const
```

DESCRIPTION

Update informs Shore of an application's intent to modify a Shore object. In particular, it obtains an EX (exclusive-mode) lock on the object if such a lock is not already held by the current transaction. Update must be called before any non-"const" method of the object is called. If the application fails to call **update** on an object before each call of a non-"const" method, the compiler will flag the method call as an error.

```
Ref<myInterfaceType> obj = REF(myInterfaceType)::lookup(...

obj.update()->non_const_method1();
obj.update()->non_const_method2();
```

An alternative, and ever-so-slightly more efficient way to use **update** is to use non-const references, which call **update** behind the scenes once upon assignment, and can thereafter be used to call non-const references.

```
WRef<myInterfaceType> obj = WREF(myInterfaceType)::lookup(...
// update has already been called.

obj.non_const_method1();
obj.non_const_method2();
```

If either **update** or the copy operator for WRef<T> encounters any errors, it calls the installed error handler. Applications can reduce the possibility of such errors by using Ref<T>::fetch (see **fetch(cxxlb)**), Ref<T>::valid **valid(cxxlb)**), or **access(oc)** to obtain an exclusive lock before calling **update**.

VERSION

This manual page applies to Version 1.1 of the Shore software.

SPONSORSHIP

The Shore project is sponsored by the Advanced Research Project Agency, ARPA order number 018 (formerly 8230), monitored by the U.S. Army Research Laboratory under contract DAAB07-91-C-Q518.

COPYRIGHT

Copyright © 1994, 1995, 1996, 1997, Computer Sciences Department, University of WisconsinMadison. All Rights Reserved.

SEE ALSO

access(oc), **fetch(cxxlb)**, **method(cxxlb)**, **ref(cxxlb)**, **valid(cxxlb)**.