

NAME

REF – pointer to Shore object

SYNOPSIS

```
// macros found in <ShoreApp.h>
class Ref<T>;
class WRef<T>;

// templates equivalent to the above macros
template class Ref<T>;
template class WRef<T>;
```

DESCRIPTION

A C++ program's access to a Shore object of type *T* is through the classes Ref<T> and WRef<T>, equivalently, Ref<T> and WRef<T>.

The form Ref<T> (template class Ref<T>) is for read-only access to objects. Writable access is through the writable references, WRef<T> (template class WRef<T>), which are derived from Ref<T>, and thus inherit all the methods of REF(T).

A writable reference to an object can be constructed from a read-only reference:

```
Ref<myInterfaceType> r;

Ref<T>::lookup(fname, r);
if(r) {
    // r now points to the registered
    // object with name fname
    // and you can now invoke any const
    // methods of Ref<T>
}

WRef<myInterfaceType> w(r);
// OR
WRef<myInterfaceType> w = r;
if(w) {
    // you can now invoke any non-const
    // methods of Ref<T>/WREF(T)
}
```

Creating a writable reference to an object does not cause the object to be inspected, or any of its access control information to be queried. Access to an object is checked only when the reference is followed.

These templates are generated with the SDL language binding. The public methods are describe in the manual pages listed below.

SEE ALSO**valid(cxxlb)**

The public methods of Ref<T> are described in **assign(cxxlb)**, **construct(cxxlb)**, **create(cxxlb)**, **create_xref(cxxlb)**, **destroy(cxxlb)**, **equal(cxxlb)**, **fetch(cxxlb)**, **flush(cxxlb)**, **get_lockmode(cxxlb)**, **get_lroid(cxxlb)**, **get_type(cxxlb)**, **is_resident(cxxlb)**, **lookup(cxxlb)**, **method(cxxlb)**, **update(cxxlb)** and **valid(cxxlb)**.

For information about the SDL compiler, see **sdl(sdl)**.

For information about the SDL C++ language binding generator, see **sdlcxx(sdl)**.