Object Additions

Category Of: Object

Declared In: remote/transport.h

Category Description

The Distributed Objects system adds two methods, **isProxy** and **encodeRemotelyFor: freeAfterEncoding:isBycopy:**, to the root Object class. These methods allow all normal objects to be remotely accessed and allow objects to be differentiated from proxies acting in their stead. Only these two method are described here. See Chapter 1, "Root Class," for a general description of the Object class and the methods it defines.

Instance Methods

encodeRemotelyFor:freeAfterEncoding:isBycopy:

– **encodeRemotelyFor:**(NXConnection *)*connection*
freeAfterEncoding:(BOOL *)*flagp*
isBycopy:(BOOL)*isBycopy*

Encodes a proxy for the receiving object over the supplied connection to ensure that all objects are capable of being remotely accessed.

This method is responsible for returning the object that must be encoded to send the receiver over *connection*. The default implementation returns a local proxy to the receiver which, when encoded, yields a remote proxy that forwards all messages to the original object.

You can override this method to change how an object is transported. If you return another object (like **self**), that object will be encoded instead. The returned object must conform to the **NXTransport** protocol. You may wish to test the *isBycopy* flag and return **self** only if the object (rather than a proxy) is to be copied across the connection. If you want the receiving object to be freed after it is encoded, you can set the boolean pointed to by *flagp* to YES.

isProxy

– (BOOL)**isProxy**

Returns NO to indicate that the receiver is a normal object and not a proxy. This method is also implemented by the NXProxy class (where it returns YES), so you can send this message to any object to determine whether it is a real object or a proxy.