NSPortCoder

Inherits From:	NSCoder : NSObject
Conforms To:	NSObject (NSObject)
Declared In:	Foundation/NSPortCoder.h

Class Description

NSPortCoder is a concrete subclass of NSCoder used in the distributed objects system to transmit object proxies (and sometimes objects themselves) between NSConnection objects. An NSPortCoder is always created and used by an NSConnection object; your code should never need to explicitly create or use one.

Making Substitutions During Encoding

Like its abstract superclass, NSCoder, NSPortCoder makes use of substitution methods that allow an object to encode itself as an instance of another class or to replace another object for itself. An object may need to offer a different replacement when being encoded specifically by an NSPortCoder, however, so instead of the generic **classForCoder:** and **replacementObjectForCoder:** methods, NSPortCoder invokes **classForPortCoder:** and **replacementObjectForPortCoder:**. Their default implementations in NSObject fall back to the generic methods, providing reasonable default behavior. (NSPortCoder doesn't use a special substitution method for decoding; it simply uses **awakeAfterUsingCoder:** as NSCoder does.)

The generic **classForCoder:** method is most useful for mapping private subclass hierarchies through a public superclass, which (for example) aids the stability of archives when subclasses are private or subject to change. Since processes communicating at run time typically use the same version of a class library, this mapping is often not needed in distributed objects communication. **classForPortCoder:** allows an object to bypass or override the generic **classForCoder:** behavior, sending its real class (or simply a different one from the generic case) to the communicating process or thread. If you implement a group of classes that use the generic **classForCoder:** method, you should also consider using **classForPortCoder:** to handle the special case of encoding with the distributed objects system.

The generic **replacementObjectForCoder:** method offers a standard way to substitute a different instance at encoding time. **replacementObjectForPortCoder:** specifically allows for the substitution of proxies over a distributed objects connection. The receiver of a **replacementObjectForPortCoder:** message can ask the NSPortCoder whether it should be encoded **bycopy** or not, and return itself or a proxy as appropriate. NSObject's implementation always returns a proxy, so subclasses that allow **bycopy** encoding should override **replacementObjectForPortCoder:** to perform at least as this sample does:

```
- (id)replacementObjectForPortCoder:(NSPortCoder *)encoder
{
    if ([encoder isBycopy]) return self;
    return [super replacementObjectForPortCoder:encoder];
}
```

If the NSPortCoder returns YES when sent an **isBycopy** message, this example method returns **self**, which will result in the receiver being sent an **encodeWithCoder:** message. If the NSPortCoder returns NO, this method invokes the superclass's implementation, which typically returns an NSDistantObject.

Method Types

Getting the NSConnection	– connection
Encoding NSPort objects	– encodePortObject: – decodePortObject
Checking for **bycopy** encoding	– isBycopy

Instance Methods

connection

– (NSConnection *)**connection**

Returns the NSConnection object that uses the NSPortCoder. In an object's **encodeWithCoder:** method, this is the sending (server) NSConnection. In **initWithCoder:** this is the receiving (client) NSConnection.

decodePortObject

– (NSPort *)**decodePortObject**

Decodes and returns an NSPort object that was previously encoded with any of the general **encode...Object:** messages. This method is primarily for use by NSPort objects themselves; you can always use **decodeObject** to decode any object.

NSPort invokes this method in its **initWithCoder:** method so that the appropriate kernel information for the port can be decoded. A subclass of NSPortCoder shouldn't decode an NSPort by sending it an **initWithCoder:** message. See "Creating a Subclass of NSCoder" in the NSCoder class specification for more informaton.

encodePortObject

– (void)**encodePortObject:**(NSPort *)*aPort*

Encodes *aPort* so that it can be properly reconstituted in the receiving process or thread. This method is primarily for use by NSPort objects themselves; you can always use the general **encode...Object:** methods to encode any object.

NSPort invokes this method in its **encodeWithCoder:** method so that the appropriate kernel information for the port can be encoded. A subclass of NSPortCoder shouldn’t encode an NSPort by sending it an **encodeWithCoder:** message. See ““Creating a Subclass of NSCoder”” in the NSCoder class specification for more informaton.

isBycopy

– (BOOL)**isBycopy**

Returns YES if the NSPortCoder is encoding an object **bycopy**, NO if it expects a proxy. You typically use this method in **replacementObjectForPortCoder:** to decide whether to substitute a proxy. See ““Making Substitutions During Encoding”” in the class description for more information.