NXEncoding

Adopted By: a private class

Declared In: remote/transport.h

Protocol Description

An object that implements the NXEncoding protocol is passed as the *portal* argument for the **encodeUsing:** message to distribute an object that adopts the NXTransport protocol. The object implementing the **encodeUsing:** method should send the *portal* object messages from the NXEncoding protocol to encode the data required to instantiate a copy of the object on the other end of the connection.

Every method in the NXEncoding protocol has a corresponding method in the NXDecoding protocol that will be used to receive encoded data. See the Distributed Objects introduction for more information.

Instance Methods

encodeBytes:count:

– **encodeBytes:**(const void *)*buffer* **count:**(int)*count*

Encodes the buffer (of size *count* bytes) indicated by *buffer.*

encodeData:ofType:

– **encodeData:**(void *)*data* **ofType:**(const char *)*type*

Encodes the data structure pointed to by *data*, whose fields are indicated by the character string *type*, consisting of the following values:

Format Character	Data Type
c	char
s	short
i	int
f	float
d	double
@	id
*	char *
%	NXAtom
:	SEL
!	int; corresponding data won't be read or written
{<type>}	struct
[<count><type>]	array

encodeMachPort:

– **encodeMachPort:**(port_t)*port*

Encodes the Mach port *port*.

encodeObject:

– **encodeObject:***anObject*

Usually encodes a proxy to *anObject*. The object to be encoded is determined by sending *anObject* an **encodeRemotelyFor:freeAfterEncoding:isBycopy:** message, which will, by default, return a proxy to *anObject*.

encodeObjectBycopy:

– **encodeObjectBycopy:***anObject*

Usually encodes *anObject*, so that a copy will be instantiated on the other end of the connection; the object to be encoded is determined by sending *anObject* an **encodeRemotelyFor:freeAfterEncoding:isBycopy:** message. *anObject* must conform to the NXTransport protocol.

encodeVM:count:

– **encodeVM:**(const void *)*bytes* **count:**(int)*count*

Encodes memory (of *count* bytes) that was allocated with **vm_allocate()**.