# IXTransientMessaging

**Adopted By:**       IXRecordManager

**Declared In:**       indexing/protocols.h

## Protocol Description

The IXTransientMessaging protocol defines a set of methods for retrieving the return values of methods from records stored in a repository (the methods must take no arguments).   It's useful for getting these values without your code having to explicitly instantiate the stored records.   All of the methods perform type casting, if necessary; for example, it's meaningful to request the string value of a floating point return value.

Since the records addressed by these methods are actually activated and messaged, you should be wary of any side effects that may be triggered.   The implementor of this protocol frees the unarchived record without re-archiving it, so changes to the state of the unarchived record won't be retained.

## Instance Methods

### getDoubleValue:ofMessage:forRecord:

- (BOOL)**getDoubleValue:**(double *)*aValue*
**ofMessage:**(SEL)*aSelector*
**forRecord:**(unsigned int)*aHandle*

Returns by reference the value that *aSelector* returns when sent to the record identified by *aHandle*. The return value of *aSelector* is interpreted as a double-precision floating-point number.   Returns YES if the value is successfully retrieved, NO otherwise.

### getFloatValue:ofMessage:forRecord:

- (BOOL)**getFloatValue:**(float *)*aValue*
**ofMessage:**(SEL)*aSelector*
**forRecord:**(unsigned int)*aHandle*

Returns by reference the value that *aSelector* returns when sent to the record identified by *aHandle*. The return value of *aSelector* is interpreted as a double precision floating point number.   Returns YES if the value is successfully retrieved, NO otherwise.

### getIntValue:ofMessage:forRecord:

- (BOOL)**getIntValue:**(int *)*aValue*
**ofMessage:**(SEL)*aSelector*
**forRecord:**(unsigned int)*aHandle*

Returns by reference the value that *aSelector* returns when sent to the record identified by *aHandle*. The return value of *aSelector* is interpreted as an integer.   Returns YES if the value is successfully retrieved, NO otherwise.

### getObjectValue:ofMessage:forRecord:

   **-** (BOOL)**getObjectValue:**(Object **)*aValue*
**ofMessage:**(SEL)*aSelector*
**forRecord:**(unsigned int)*aHandle*

Returns by reference the value that *aSelector* returns when sent to the record identified by *aHandle*.
The return value of *aSelector* is interpreted as an Objective C object.   Returns YES if the value is
successfully retrieved, NO otherwise.

### getOpaqueValue:ofMessage:forRecord:

   **-** (BOOL)**getOpaqueValue:**(NXData **)*aValue*
**ofMessage:**(SEL)*aSelector*
**forRecord:**(unsigned int)*aHandle*

Returns by reference the value that *aSelector* returns when sent to the record identified by *aHandle*.
The return value of *aSelector* is interpreted as an instance of class NXData.   Returns YES if the
value is successfully retrieved, NO otherwise.

### getStringValue:ofMessage:forRecord:

   **-** (BOOL)**getStringValue:**(char **)*aValue*
**ofMessage:**(SEL)*aSelector*
**forRecord:**(unsigned int)*aHandle*

Returns by reference the value that *aSelector* returns when sent to the record identified by *aHandle*.
The return value of *aSelector* is interpreted as a string.   Returns YES if the value is successfully
retrieved, NO otherwise.   The sender is responsible for freeing the string.

### getStringValue:inLength:ofMessage:forRecord:

   **-** (BOOL)**getStringValue:**(char **)*aValue*
**inLength:**(unsigned int)*aLength*
**ofMessage:**(SEL)*aSelector*
**forRecord:**(unsigned int)*aHandle*

Returns by reference the value that *aSelector* returns when sent to the record identified by *aHandle*.
The return value of *aSelector* is returned by copy as a string in the supplied buffer,   of length up to
*aLength*.   Returns YES if the value is successfully retrieved, NO otherwise.