

# IXTransientAccess

**Adopted By:** IXRecordManager

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

## Protocol Description

The IXTransientAccess protocol defines a set of methods for retrieving the values of the instance variables of records stored in a repository. It's useful for getting these values without explicitly instantiating the stored records. All of the methods perform type casting, if necessary; for example, it's meaningful to request the integer value of a floating-point instance variable.

## Instance Methods

### **getDoubleValue:ofIvar:forRecord:**

- (BOOL)**getDoubleValue:(double \*)aValue**  
**ofIvar:(const char \*)ivarName**  
**forRecord:(unsigned int)aHandle**

Returns by reference the value of the instance variable *ivarName* from the record identified by *aHandle*. The value of *ivarName* is returned as a double-precision floating-point number. Returns YES if the value is successfully retrieved, NO otherwise.

### **getFloatValue:ofIvar:forRecord:**

- (BOOL)**getFloatValue:(float \*)aValue**  
**ofIvar:(const char \*)ivarName**  
**forRecord:(unsigned int)aHandle**

Returns by reference the value of the instance variable *ivarName* from the record identified by *aHandle*. The value of *ivarName* is returned as a single-precision floating-point number. Returns YES if the value is successfully retrieved, NO otherwise.

### **getIntValue:ofIvar:forRecord:**

- (BOOL)**getIntValue:(int \*)aValue**  
**ofIvar:(const char \*)ivarName**  
**forRecord:(unsigned int)aHandle**

Returns by reference the value of the instance variable *ivarName* from the record identified by *aHandle*. The value of *ivarName* is returned as an integer. Returns YES if the value is successfully retrieved, NO otherwise.

### **getObjectValue:ofIvar:forRecord:**

- (BOOL)**getObjectValue:(Object \*\*)aValue**  
**ofIvar:(const char \*)ivarName**  
**forRecord:(unsigned int)aHandle**

Returns by reference the value of the instance variable *ivarName* from the record identified by

*aHandle*. The value of *ivarName* is returned as an Objective C object. Returns YES if the value is successfully retrieved, NO otherwise.

**getOpaqueValue:ofIvar:forRecord:**

- (BOOL)getOpaqueValue:(NXData \*\*)aValue

ofIvar:(const char \*)ivarName

forRecord:(unsigned int)aHandle

Returns by reference the value of the instance variable *ivarName* from the record identified by *aHandle*. The value of *ivarName* is returned as an instance of class NXData. Returns YES if the value is successfully retrieved, NO otherwise.

**getStringValue:ofIvar:forRecord:**

- (BOOL)getStringValue:(char \*\*)aValue

ofIvar:(const char \*)ivarName

forRecord:(unsigned int)aHandle

Returns by reference the value of the instance variable *ivarName* from the record identified by *aHandle*. The value of *ivarName* is returned as a string. Returns YES if the value is successfully retrieved, NO otherwise. The sender is responsible for freeing the string.

**getStringValue:inLength:ofIvar:forRecord:**

- (BOOL)getStringValue:(char \*\*)aValue

inLength:(unsigned int)aLength

ofIvar:(const char \*)ivarName

forRecord:(unsigned int)aHandle

Returns by reference the value of the instance variable *ivarName* from the record identified by *aHandle*. The value of *ivarName* is returned by copy as a string in the supplied buffer, truncated if needed to fit within *length* bytes (including the terminating null character). Returns YES if the value is successfully retrieved, NO otherwise.