

```
writeBuffer:(void **)bufferPtr  
ofLength:(unsigned *)lengthPtr  
usingFormat:(const char *)aFormatName
```

If your application creates a custom class that's associated with a property and your class implements this method, this method will be invoked automatically when the Database Kit tries to store the object into the database.

The pointer `*bufferPtr` may point either to an existing buffer or to `NULL`. If it points to an existing buffer, `*lengthPtr` points to the buffer's length. But if `*bufferPtr` points to `NULL`, the method should allocate space for a new buffer as needed, and should write the length it allocates into the location that `*lengthPtr` points to.

The argument `aFormatName` is a string containing the name of the format in which the data is written. The Database Kit defines the following names for formats:

```
.aEPSo  
.aRTFo  
.aTIFFo
```

Alternatively, the string may be the name of any type your application recognizes, as declared in DBModeler's Attribute Inspector for data of type Object.