

Q: When I use the DBBinder evaluateString method to issue SELECT statements with an Oracle adaptor and retrieve data from my tables, the LONG column values are always returned as NULL. What is going wrong?

A: This is a known problem with the Oracle Adaptor in Releases 3.1 and 3.2. If you use InterfaceBuilder and the DBModule fetch mechanism to retrieve data from LONG columns, it works properly. However, if you must use a binder object, you need to set up your property list before calling the binder evaluateString. For example, if you have a table **A** with two columns **attr1** and **attr2**, where **attr1** is defined as integer, and **attr2** is defined as LONG, you need to create a property list that includes these two attributes, and initialize the binder accordingly. See the code snippet below as an example:

```
/* Initialize the binder and set it up */
```

```
DBBinder *binder = [[DBBinder alloc] init];  
List *resultList = [[List alloc] init];  
List *props = [[List alloc]init];
```

```
[binder setDatabase:myDatabase];  
[binder setContainer:(id)resultList];
```

/\* This is the workaround to the LONG datatype problem.

You must create a property list that contains all the columns you want to select in your query before doing the evaluateString. \*/

```
myTable = [myDatabase entityNamed:"A"];  
[props addObject:[myTable propertyNamed:"attr1"]];  
[props addObject:[fooTable propertyNamed:"attr2"]];  
[binder setProperties:props];  
result = [binder evaluateString:"select * from A"];
```

/\* Do the rest of the work, such as fetching data,  
freeing the binder and other resources after they are used, etc...\*/

QA894

Valid for 3.1, 3.2