

# EOF-Modeling-Common Pitfalls and Troubleshooting

## Value Types

In EOModeler, the Value Type for an attribute can be set while editing an attribute in "Table Mode". If the "Value Type" column is not visible, use the pop-up menu labeled "Add Column" in the lower left corner of the pane. The Value Type of an attribute in an EOModel controls how the JDBC adaptor handles important details in its negotiations with the database. A Value Type is typically a single character.

For attributes with a Value Class of `java.lang.Number` the following value types are defined:

```
b = Byte
s = Short
i = Integer
l = Long
f = Float
d = Double
B = java.math.BigDecimal
c = Boolean
```

Since `java.lang.Number` is an abstract superclass, the Value Type controls which concrete class the JDBC adaptor should instantiate based on the raw data provided from the database.

It also controls which JDBC methods are used to send and retrieve the data to and from the database. For attributes with a Value Class of `java.lang.String` the following value types are defined:

```
<none> = Backwards compatible: Selects S or C option automatically
c = Trimmed for use with fixed length CHAR columns
S = String, will fail when string length is too large
C = Character stream (for large strings)
E = Binary stream (not recommended)
```

These Value Types affect which method on the `java.sql.PreparedStatement` are used to transfer text data between the database and the JDBC adaptor. An empty Value Type is backward compatible behavior with WebObjects v4.5 and uses the `setString()` method if the text is less than the database's advertised maximum varchar length, and `setCharacterStream()`, if it is too large. If the database fails to advertise a maximum length, the default is 256 characters. A Value Type of 'S' uses `setString()` regardless of the text's length. A Value Type of 'C' uses `setCharacterStream()` regardless of the text's length. A Value Type of 'E' converts the text into raw UTF-8 bytes, and then uses `setBinaryStream()` to save them in a binary typed column in the database. The Value Type of 'c' tells the adaptor to generate SQL using `RTRIM` to strip off all trailing spaces, such as those found in CHAR columns.

'S' is appropriate for most text columns. 'C' is good for columns which usually contain large amounts of data. 'c' should be used when trailing spaces are not significant in a CHAR column. (It's better to use a VARCHAR column if possible.) We recommend against using 'E', except in extreme cases. It is the database's responsibility to handle text encoding issues, and using 'E' is usually an indication the database is not correctly configured.

For attributes with a Value Class of `NSTimestamp` the following value types are defined:

```
<none> = get/setObject
D = get/setDate
t = get/setTime
T = get/setTimeStamp
M = bug workaround for MSSQLServer JDBC
```

These value types affect how the data is transfer between the JDBC adaptor and the database. An empty Value Type uses `get/setObject()` on the `ResultSet` and `PreparedStatement`. It assumes the database can provide a value compatible with a `java.sql.Timestamp`. A 'D' indicates WebObject's JDBC adaptor should use `get/setDate`. A 't' indicates `get/setTime()`, and a 'T' uses `get/setTimeStamp()`. The 'M' value type is a workaround for a bug in some MS SQLServer JDBC drivers. It only support `java.sql.Date`.

BLOB and CLOB columns are handled specially by the adaptor to support Oracle. The Value Type has no influence.

Source: [http://developer.apple.com/documentation/LegacyTechnologies/WebObjects/WebObjects\\_5.1/ReleaseNotes/ReleaseNotes.html](http://developer.apple.com/documentation/LegacyTechnologies/WebObjects/WebObjects_5.1/ReleaseNotes/ReleaseNotes.html)

## Row Locking

EO uses a system of row locking where every UPDATE into the database is accompanied by a WHERE clause that contains the expected original values of the data. That way if the data changes between the initial FETCH and the subsequent UPDATE, the UPDATE fails and EO can throw an exception. The set of fields to be used during row locking can be specified in the Modeler application.

If a String field is one of the fields used for row locking and the value type is 'C' (Character Stream) and the data of the field is larger than 255, the UPDATE fails. This seems like a bug.

The failure can lead to errors such as 'com.webobjects.eoaccess.EOGeneralAdaptorException: deleteRowDescribedByQualifierEntity - com.webobjects.jdbcadaptor.JDBCChannel: method failed to delete row in database'.