

EOF-Using EOF-Common Pitfalls and Troubleshooting

Deadlocks

The most common cause of this is `EOEditingContextLock` but you can also cause problems by leaving locks on the `EODatabaseContext`. To find out where you are locking improperly you will need to start tracking editing context locks. Review the [Context and Database Locking](#) section for more information.

Subclass Missing Primary Key

If you are subclassing across EO models and EOF is not generating a primary key in SQL when you insert your subclass (the result would be an integrity violation for a null primary key), then you have likely run into the annoying behavior of EOF when dealing with models that have SLIGHTLY different connection dictionaries. Basically, EOF decides when to share EODatabases across multiple models by comparing the connection dictionaries. However, the connection dictionary contains a large block named `jdbc2Info` with cached type and driver information. If you happened to create your models with, for instance, different jdbc driver versions, that version number will cause the dictionaries to not be equals. When this occurs, EOF assigns each model to its own EODatabase, which causes SQL generation for the subclass entity to enter a codepath that ends up not generating a primary key. Chuck Hill recommends removing `jdbc2Info` at runtime to prevent problems like this from occurring.

No Suitable Driver Found

If you get the following exception:

```
com.webobjects.jdbcadaptor.JDBCAdaptorException: No suitable driver
    at
com.webobjects.jdbcadaptor.JDBCContext.connect(JDBCContext.java:244)
    at
com.webobjects.jdbcadaptor.JDBCContext._tryConnect(JDBCContext.java:331)
    at
com.webobjects.jdbcadaptor.JDBCContext._channelWillOpen(JDBCContext.java:4
68)
    at
com.webobjects.jdbcadaptor.JDBCChannel.openChannel(JDBCChannel.java:81)
```

Odds are very good that your classpath is incorrect. There are two parts to database support in EOF, the JDBC driver, and the Database Plugin. You must make sure that BOTH are in your classpath. For instance, on FrontBase, you must have both `frontbasejdbc.jar` AND `frontbaseplugin.jar` (from the `FrontBasePlugin.framework`) in your classpath.