

Explanation of the default properties in a Wonder project

When you create a new Wonder project in WOLips, the Properties file is filled with numerous properties. If you are new to WebObjects or new to using Project Wonder, you might want to know what those properties do. Let's see:

```
# OGNL
ognl.helperFunctions=true
ognl.inlineBindings=true
```

The OGNL enable you to use the OGNL syntax for inline bindings (eg, putting bindings directly in the .html file instead of the .wod, for example: `<wo:str value="$pageTitle" />`). WOOGNL is automatically added to the classpath when you create a new Wonder project.

```
# Misc
er.extensions.stackTrace.cleanup=true
```

See `ERXExceptionUtilities.printStackTrace`

```
# EOF
er.extensions.ERXEC.safeLocking=true
```

This one is to enable editing context auto-locking (eg, you don't have to lock/unlock it yourself) at the end of the RR loop. You have to create your editing context with `ERXEC.newEditingContext`.

```
er.extensions.ERXEC.useSharedEditingContext=false
```

If set to true, newly created instances use the shared editing context.

```
er.extensions.ERXEnterpriseObject.applyRestrictingQualifierOnInsert=true
```

From what I understand, the "Qualifier" field defined in the model for an entity will not be used if you set this property to false.

```
er.extensions.ERXEnterpriseObject.updateInverseRelationships=true
```

Provides automatic inverse relationship updating for `ERXGenericRecord` and `ERXCustomObject`.

When to put it at FALSE

If you have a mix of `ERXGenericRecords` and `EOGenericRecords` between models, you're better of turning it off.

Since an `EOGenericRecords` doesn't have the facilities to update the inverse relationship, you will get a `ClassCastException`.

```
#dbConnectURLGLOBAL=  
#dbConnectUserGLOBAL=  
#dbConnectPasswordGLOBAL=
```

Those properties, if used, allow you to specify the JDBC/EOF information for all of your models. See [Best Practices-Properties Files](#)

```
# Migrations  
#er.migration.migrateAtStartup=true  
#er.migration.createTablesIfNecessary=true  
#er.migration.modelNames=YourModel
```

Those properties are to enable support for [migrations](#). If you are using migrations but you have models where the schema already exist in the database, uncomment `er.migration.modelNames` and put in a list (comma separated) of the models you want to use migrations with, if you let this property commented, Wonder will try to migrate all of your models, which would generate errors for models where your schema already exist.

```
# Misc Components  
#er.extensions.ERXWOForm.multipleSubmitDefault=true
```

Set the "multipleSubmit" binding of WOForm to true for all WOForm instances. Uncomment that property if you don't want to remember to set that dreaded binding on every form.

```
#er.extensions.ERXWORepetition.checkHashCodes=true
```

Help with backtracking issues by adding not only the current index, but also the current object's hash code to the element id (see [ERXWORepetition API](#)).

```
#er.extensions.ERXWORepetition.raiseOnUnmatchedObject=true
```

If this property is set to true, an `ERXWORepetition.UnmatchedObjectException` is thrown if `notFoundMarker` is bound, that is used for the item in the repetition. (see [ERXWORepetition API](#)).

```
#er.extensions.ERXWORepetition.eoSupport=true
```

Try to use `globalIDs` to determine the `hashCode` for EOs.

```
#er.component.clickToOpen=true  
#wolips.password=wo
```

Enable support for [Click to Open](#) support in WOLips.

```
# ERJavaMail
er.javamail.centralize=true
er.javamail.adminEmail=admin@yourhost.com
er.javamail.debugEnabled=false
er.javamail.smtpHost=your.smtpserver.com
er.javamail.smtpAuth=false
er.javamail.smtpUser=
er.javamail.smtpPassword=
```

Check [Project WONDER-Frameworks-ERJavaMail](#) for an explanation.

```
# Log4j Configuration
log4j.loggerFactory=er.extensions.logging.ERXLogger$Factory
log4j.rootCategory=INFO,A1
log4j.appender.A1=er.extensions.logging.ERXConsoleAppender
log4j.appender.A1.layout=er.extensions.logging.ERXPatternLayout
log4j.appender.A1.layout.ConversionPattern=%d{MMM dd HH:mm:ss} %$[%#] %-5p
%c %x - %m%n

# Log4j Categories
# Here are a few log4j sub-categories that are interesting.
# Don't forget that in development mode this file will get reloaded
# everytime it changes, so if you say want to turn adaptor debugging
# on in the middle of the app simply set the below category to debug.
# Very handy.
# Base Category
log4j.logger.er=INFO
```

Standard Log4J stuff. Log4J is bundled into ERJars, and make it sure that you don't have a copy of it in /Library/Java/Extensions or it will conflict with the one from ERJars, especially if the one in /Library/Java/Extensions is older than the one in ERJars.

```
# ERExtensions
# Transaction - Switching this to debug will start the sql ouputting.
log4j.logger.er.transaction.adaptor.EOAdaptorDebugEnabled=DEBUG
```

With this set to DEBUG, all SQL stuff sent to your RDMS will show up in the console (or where you tell Log4J to send his stuff).

```
# Fixes - Turning this on will show all the models that are loaded
log4j.logger.er.extensions.fixes.ERSharedEOLoader=INFO
```

See ERXSharedEOLoader.java in ERExtensions.