

EOModelDoc

Contents

- [#Overview](#)
- [#Future Plans](#)
- [#Download and Example Output](#)
- [#Usage](#)
- [#Running from Ant](#)
- [#Template Files](#)
- [#Alternate Templates](#)

Overview

EOModelDoc is a command line tool to generate HTML documentation of a set of EOModels. It is a replacement for [EORporter](#) (which will not work on Leopard and newer releases of OS X). EOModelDoc was funded by the Apple iTunes Store and released to the community.

On the WOLips side, from version 3.3.x, [Entity Modeler](#) has a Documentation tab on all of the elements of a model that lets you record HTML documentation for your model, entities, attributes, etc. There is even an HTML preview. You can then run eomodeldoc to produce documentation much like javadoc for your model.

Future Plans

From Mike Schrag: *There's not a nice build process for this one either, yet, because it includes all of the eclipse dependencies in the single jar. Eventually I'll make this nicer and include it actually inside of Eclipse. There are also modifications to the ERNeutralLook coming soon that will display popup help for you models based on the model documentation.*

Download and Example Output

[Download EOModelDoc jar](#)

This is an example of the current state of eomodeldoc (the default templates are not done, but it will give you an idea of what it can do). The templates are all Velocity-based, and you can override them with custom versions. This was run in the BugTracker project folder. This model does not have documentation added to it with the new Entity Modeler yet, so it looks a little more bare than it would if it was documented properly. I'll probably document it just to have a full example to show.

[View Example Output](#)

Usage

The command line syntax of for EOModelDoc:

```
java -jar eomodeldoc.jar -output /path/to/output/folder [-model /path/to/model.eomodeld]* [-modelgroup /path/to/working/dir] [-templates /path/to/templates] [-entityURLTemplate "http://domain/path/to/JavaDocs/$entity.classNamePath.html]
```

Parameter	Usage / Function
output	Path to the directory where the output HTML files will be written.
model	Path to one model to generate documentation for. This parameter can be used multiple times, once for each model.
modelgroup	The path to an Eclipse or IDEA project. All models listed in the project will be documented. This defaults to the current working directory.
templates	Path to the directory where the Velocity templates are stored. See the #Template Files section below for details on these templates. If this is not specified, the default templates inside the eomodeldoc.jar file will be used.
entityURLTemplate	URL to the JavaDocs for the classes in the model. This is used to create links from the EOModel documentation to the JavaDoc documentation. The dots in the classpath are replaced with "/" when the path is generated.

Notes

- Only `-output` is required
- Usually only one of `-modelGroup` or `-model` is used (i.e. not both at the same time)
- The `-templates` path only needs to point to the templates that you have changed. If any templates are not found on that path, the defaults from the jar are used.
- For `-entityURLTemplate` the `http://domain/path/to/JavaDocs/` the path to the index.html file that JavaDoc generates. It will be something like `http://developer.apple.com/documentation/MacOSXServer/Reference/W054_Reference/`. If the EOModelDocs and JavaDocs are on the same server, you can use absolute or relative paths instead of a full URL:`http://developer.apple.com/documentation/MacOSXServer/Reference/W054_Reference/`

Usage Examples

Document one model:

```
java -jar /path/to/eomodeldoc.jar -model /path/to/model.eomodeld -output /tmp/eomodeldoc
```

Document all models based on the Eclipse or IDEA project definition:

```
java -jar /path/to/eomodeldoc.jar -modelgroup /path/to/project -output /tmp/eomodeldoc
```

Document all models based on the Eclipse/IDEA project definition (when current working directory is the top level of the project):

```
java -jar /path/to/eomodeldoc.jar -output /tmp/eomodeldoc
```

Running from Ant

Note: do not put this jar on your classpath! Chaos and Other Bad Things may ensue.

There is no Ant task (yet) for EOModelDoc. You can run it with an `java` task following this pattern:

```
<java jar="/path/to/eomodeldoc.jar"
  fork="true"
  maxmemory="256m">
  <arg value="-output" />
  <arg value="/path/to/output/folder" />
  <arg value="-modelgroup" />
  <arg value="/path/to/working/dir" />
  <arg value="-templates" />
  <arg value="/path/to/templates" />
  <arg value="-entityURLTemplate" />
  <arg value="http://whatever?${entity.classNamePath}.html" />
</java>
```

```

<java jar="/path/to/eomodeldoc.jar"
      fork="true"
      maxmemory="256m">
  <arg value="-output" />
  <arg value="/path/to/output/folder" />
  <arg value="-model" />
  <arg value="/path/to/modelA.eomodeld" />
  <arg value="-model" />
  <arg value="/path/to/modelB.eomodeld" />
  <arg value="-model" />
  <arg value="/path/to/modelC.eomodeld" />
  <arg value="-templates" />
  <arg value="/path/to/templates" />
  <arg value="-entityURLTemplate" />
  <arg value="http://whatever?${entity.classNamePath}.html" />
</java>

```

If you include this in the build.xml file in the project, it can be as simple as:

```

<java jar="/path/to/eomodeldoc.jar"
      fork="true"
      maxmemory="256m">
  <arg value="-output" />
  <arg value="Documentation" />
  <arg value="-modelgroup" />
  <arg value="." />
</java>

```

Template Files

Default templates are included in the eomodeldoc.jar file. You can extract these templates to use as a basis for your own templates with this command:

```
unzip /path/to/eomodeldoc.jar *.vm -x "*/*"
```

See the [EOGenerator template documentation](#) documentation for information on customizing these templates. The `-templates` path only needs to point to the templates that you have changed. If any templates are not found on that path, the defaults from the jar are used.

The templates and their function are:

Template Name	Usage / Function
indexModels.html.vm	Left pane index of all models.
indexOverview.html.vm	Left pane index of the contents of all models (e.g. list of entities and stored procedures).

modelOverview.html.vm	Left pane index of the contents of one model (e.g. list of entities and stored procedures).
indexContent.html.vm	Main frame description of each model and contents of each model (e.g. list of entities and stored procedures).
modelContent.html.vm	Main frame contents of one model (list of entities and stored procedures).
entityContent.html.vm	Main frame contents for one Entity
storedProcedureContent.html.vm	Main frame contents for one stored procedure.
index.html.vm	Top level index.html file defining frameset. You probably won't need to modify this.
prototype.js.vm	Prototype JavaScript file so that the generated documentation can use Ajax whizziness. You probably won't need to modify this.
eomodeldoc.css.vm	CSS file that controls what the generate output looks like.
eomodeldoc.js.vm	JavaScript file that cleans up the display. You probably won't need to modify this.

Some pictures will illustrate this:

Models [indexModels.html](#)

- All Entities
- BugTracker
- ERAttachment
- ERCoreBusinessLogic
- ERMail
- ERTaggable

All Entities [indexOverview.html](#)

- Bug
- Comment
- Component
- Difficulty
- ERAttachment
- ERAttachmentData
- ERCAuditBlob
- ERCAuditTrail
- ERCAuditTrailEntry
- ERCHelpText
- ERLogEntry
- ERCMailMessage
- ERCMailMessageArchive
- ERCMessageAttachment
- ERCPreference
- ERCStatic
- ERDatabaseAttachment
- ERFileAttachment
- ERS3Attachment
- ERTag
- Framework
- People
- Priority
- Release
- Requirement

EOModelDoc [indexContent.html](#)

Models	
BugTracker	No documentation available for BugTracker. Bug, Comment, Component, Difficulty, Framework, People, Priority, Release, Requirement, RequirementSubType, RequirementType, TestItem, TestItemState, XBugTestItem, XCommentAttachment, XRequirementTestItem,
ERAttachment	No documentation available for ERAttachment. ERAttachment, ERAttachmentData, ERDatabaseAttachment, ERFileAttachment, ERS3Attachment,
ERCoreBusinessLogic	No documentation available for ERCoreBusinessLogic. ERCAuditBlob, ERCAuditTrail, ERCAuditTrailEntry, ERCHelpText, ERLogEntry, ERCPreference, ERCStatic,
ERMail	No documentation available for ERMail. ERCMailMessage, ERCMailMessageArchive, ERCMessageAttachment,
ERTaggable	No documentation available for ERTaggable. ERTag,

Models

All Entities
BugTracker
ERAttachment
ERCoreBusinessLogic
ERMail
ERTaggable

BugTracker

[modelOverview.html](#)

Entities

Bug
Comment
Component
Difficulty
Framework
People
Priority
Release
Requirement
RequirementSubType
RequirementType
TestItem
TestItemState
XBugTestItem
XCommentAttachment
XRequirementTestItem

BugTracker [modelContent.html](#)

All Models

Entities

Bug	No documentation available for Bug.
Comment	No documentation available for Comment.
Component	No documentation available for Component.
Difficulty	No documentation available for Difficulty.
Framework	No documentation available for Framework.
People	No documentation available for People.
Priority	No documentation available for Priority.
Release	No documentation available for Release.
Requirement	No documentation available for Requirement.
RequirementSubType	No documentation available for RequirementSubType.
RequirementType	No documentation available for RequirementType.
TestItem	No documentation available for TestItem.
TestItemState	No documentation available for TestItemState.
XBugTestItem	No documentation available for XBugTestItem.
XCommentAttachment	No documentation available for XCommentAttachment.
XRequirementTestItem	No documentation available for XRequirementTestItem.

Models

All Entities
BugTracker
ERAttachment
ERCoreBusinessLogic
ERMail
ERTaggable

BugTracker

Entities

Bug
Comment
Component
Difficulty
Framework
People
Priority
Release
Requirement
RequirementSubType
RequirementType
TestItem
TestItemState
XBugTestItem
XCommentAttachment
XRequirementTestItem

BugTracker / Bug

parent of Requirement
referenced by Comment, Component, Release, XBugTestItem
external name BUG
class name er.bugtracker.Bug

entityContent.html or storedProcedureContent.html

Attributes

show details hide non-class

componentID
COMP_ID
dateModified
DATE_MODIFIED
dateSubmitted
DATE_SUBMITTED
id [primary key]
ID
isFeatureRequest
IS_FEATURE_REQUEST
isRead
IS_READ
originatorID
ORIGINATOR_ID
ownerID
PEOPLE_ID
previousOwnerID
PREVIOUS_OWNER_ID
priorityID
PR_ID

Relationships

show details hide non-class

bugTestItems
XBugTestItem
comments
Comment
component
Component
originator
People
owner
People
previousOwner
People
priority
Priority
targetRelease
Release
testItems
TestItem via bugTestItems.testItem

Alternate Templates

Have some interesting templates to share? Add them as attachments and link them here.

Template Mark I from Chuck

Download

This is the default template with some useful (to me) changes:

- individual properties can be toggled open and closed
- non-class hidden by default
- less blank vertical space
- column name moved into details

Attributes

[show details](#) [show non-class](#)

activeStatus
The current production status, may have been modified by Pre-press from the status from Order Entry (i.e. for a returned Show). activeStatus is used to determine the priority (ColorSwatch) for an ad when it is logged or returned from a show.
activity
activityTime
column name activityTime external type date (TIMESTAMP) allows null? Yes used for locking? Yes
build
colorSide1
column name colorSide1 external type string50 (VARCHAR) allows null? No used for locking? Yes width 50
colorSide2

Relationships

[show details](#)

auditRecords
colorSwatch
comments
composer
currentActivityUser
currentStateUser
daplEntries
imageReplates
messages
order
proofer
publications
List of OrderPublications describing the Publications that this ad will appear in.