EcoreDoc User Guide

Niko Stotz

Version 0.8.0-SNAPSHOT

Table of Contents

Generates AsciiDoctor files to document Ecore metamodels, similar to JavaDoc. AsciiDoctor can be rendered as HTML, PDF, or Eclipse Help. EcoreDoc can be used as Maven Plugin, standalone command-line tool, Java API, or Eclipse Plug-in.

1. Overview

EcoreDoc's Java API works on a list of `EClassifier`s. Maven Plugin, Standalone Command-Line Tool and Eclipse Plug-in take all `EClassifier`s from one or more `*.ecore` or `*.xcore` files.

EcoreDoc creates one output document containing all passed `EClassifier`s. They are grouped by containing `EPackage`. The output document contains documentation, all properties of supported elements, and cross-references to all usages of each element.

EcoreDoc currently supports the following elements:

- `EPackage`
- `EDataType`
- `EEnum`
- `EEnumLiteral`
- `EClass`
- `EAttribute`
- `EReference`

EcoreDoc is highly configurable via the Generator Configuration.

The homepage and repository of EcoreDoc can be found at https://gitlab.manatree.io/MDEAssets/EcoreDoc. Please use the issue tracker at this site for any feature requests or bugs.

2. Java API

The Java API is available as Maven artifact `com.altran.general.emf.ecoredoc:com.altran.general.emf.ecoredoc.generator` or OSGi bundle `com.altran.general.emf.ecoredoc.generator.ebr`.
The Generator Configuration is contained in Maven artifact `com.altran.general.emf.ecoredoc:com.altran.general.emf.ecoredoc.generator.config` or OSGi bundle `com.altran.general.emf.ecoredoc.generator.config.ebr`.

The main interface is `com.altran.general.emf.ecoredoc.generator.EcoreDocGenerator`. The constructor takes the list of `EClassifier`s to generate documentation for.

The `getConfig()` method returns a fully initialized `com.altran.general.emf.ecoredoc.generator.config.EcoreDocGeneratorConfig` that can be changed to adjust the Generator Configuration.

The `generate()` method returns a `CharSequence` containing the complete AsciiDoctor output document.

3. Maven Plugin

The Maven Plugin is available as Maven artifact `com.altran.general.emf.ecoredoc:ecoredoc-maven-plugin`.

It supports the following configuration settings:

inputFiles (required)

The list of Ecore metamodel files to create documentation for.

outputFile (required)

The output file to write the generated AsciiDoctor document to.
By convention, the file extension is `.adoc`.

If the file exists, it will be overwritten and a warning is emitted.

resolve (default: false)

Whether EcoreDoc should explicitly try to resolve all references in the *inputFiles*. Might be necessary for highly interconnected metamodels.

config (default: unchanged default config)

Customized Generator Configuration.

The *config* contents stricly follow the structure and naming relative to Class EcoreDocGeneratorConfig, easiest explained with an example.

Assume the *inputFiles* contain two `EPackage`s, namely `EPackage1` and `EPackage2`.
`EPackage1` contains, among others, two `EClass`es, named `MyEClass` and `Class3`. The latter one contains, among others, the `EAttribute` named `specialNumber`.
`EPackage1` also contains an `EEnum` named `Enum1`.

pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <artifactId>my-artifact-id</artifactId>

  <dependencies>
    <dependency>
      <groupId>com.altran.general.emf.ecoredoc</groupId>
      <artifactId>ecoredoc-maven-plugin</artifactId>
    </dependency>
  </dependencies>
```

```
<build>
  <plugins>
    <plugin>
      <groupId>com.altran.general.emf.ecoredoc</groupId>
      <artifactId>ecoredoc-maven-plugin</artifactId>

      <configuration>
        <resolve>true</resolve>

        <config>
          <renderDefaults>false</renderDefaults>
          <documentTitle>This is the title of my document</documentTitle>
          <ePackages>
            <ePackage>
              <targetEPackage>EPackage1</targetEPackage>
              <eClasses>
                <eClass>
                  <targetEClass>MyEClass</targetEClass>
                  <repeatInherited>false</repeatInherited>
                </eClass>
                <eClass>
                  <targetEClass>Class3</targetEClass>
                  <eAttributes>
                    <eAttribute>
                      <targetEAttribute>specialNumber</targetEAttribute>
                      <render>false</render>
                    </eAttribute>
                  </eAttributes>
                </eClass>
              </eClasses>
              <eEnums>
                <eEnum>
                  <targetEEnum>Enum1</targetEEnum>
                  <renderDefaults>true</renderDefaults>
                </eEnum>
              </eEnums>
            </ePackage>
            <ePackage>
              <targetEPackage>EPackage2</targetEPackage>
              <renderDefaults>true</renderDefaults>
            </ePackage>
          </ePackages>
        </config>
        <inputFiles>
          <inputFile>EPackage1.ecore</inputFile>
          <inputFile>EPackage2.ecore</inputFile>
        </inputFiles>
        <outputFile>output.adoc</outputFile>
      </configuration>
    </plugin>
  </plugins>
```

```
  </build>
</project>
```

This example sets the following configuration:

- renderDefaults for all contents: `true`
- documentTitle: `This is the title of my document`
- repeatInherited for *MyEClass*: `false`
- render for *specialNumber*: `false`
- renderDefaults for *Enum1*: `true`
- renderDefaults for *EPackage2*: `true`

4. Standalone Command-Line Tool

The standalone command-line tool is available as Maven artifact `com.altran.general.emf.ecoredoc:com.altran.general.emf.ecoredoc.standalone`.

Use the following command to invoke. Please replace `${ecoredoc-version}` with your version of EcoreDoc:

```
java -jar com.altran.general.emf.ecoredoc.standalone-${ecoredoc-version}-jar-with
-dependencies.jar <options>
```

If invoked without options, it will print the following help:

```
Generates reference documentation for ecore models.

The output is inspired by JavaDoc and formatted in AsciiDoctor format.
AsciiDoctor can easily be rendered to HTML, PDF, or Eclipse help.


Usage:

EcoreDocGenerator [parameters] [List of ecore files to generate]

If unspecified, the output file name will be "<firstEcoreFile.ecore>.adoc"


Parameters:

  -r,
  --resolve: Resolve external references

  -o <outputFile>,
  --output <outputFile>: Specify output file name.
```

```
  --documentTitle <title>: Set title of output document

  --positionEDataTypes <pos>: Set rendering position of all EDataTypes within EPackage

  --positionEEnums <pos>:     Set rendering position of all EEnums within EPackage

  --positionEClasses <pos>:   Set rendering position of all EClasses within EPackage

  [+|-]defaults:   [Enable|disable] rendering of default values

  [+|-]bounds:     [Enable|disable] rendering of multiplicity bounds
                                    (overwrites defaults parameter)


  [+|-]inherited:  [Enable|disable] repetition of inherited features

  [+|-]useCases:   [Enable|disable] rendering of use cases
                                    (references to other usages of this element)


  [+|-]subTypes:   [Enable|disable] rendering of sub-types

  [+|-]superTypes: [Enable|disable] rendering of super-types


Examples:

  EcoreDocGenerator my.ecore
  Generates the documentation of my.ecore into my.ecore.adoc


  EcoreDocGenerator some/path/to/my.ecore other.ecore
  Generates the documentation of some/path/to/my.ecore and other.ecore
  into some/path/to/my.ecore.adoc


  EcoreDocGenerator -r my.ecore
  Tries to resolve all external references in my.ecore and
  generates the documentation of my.ecore and referenced models into my.ecore.adoc


  EcoreDocGenerator -defaults +bounds my.ecore
  Generates the documentation of my.ecore and referenced models into my.ecore.adoc
  without rendering default values, but still rendering multiplicity bounds


  EcoreDocGenerator --positionEClasses 1 --positionEEnums 2 --positionEDataTypes 3
my.ecore
  Generates the documentation of my.ecore and referenced models into my.ecore.adoc
  with all EClasses first, then all EEnums, and finally all EDataTypes
```

```
EcoreDocGenerator -o output.adoc my.ecore other.ecore
Generates the documentation of my.ecore and other.ecore into output.adoc
```

5. Eclipse Plug-in

The Eclipse Plug-in is available as Feature `com.altran.general.emf.ecoredoc.ui.feature`.

It provides a context menu entry for one or more `*.ecore` / `*.xcore` files in the following views:

- Project Explorer
- Package Explorer
- Model Explorer

The command creates one output file next to the first selected input file, named `<firstInputFile.ecore>.adoc`. The output file contains the documentation of all selected metamodels.

6. EcoreDoc Metamodel Annotation

Any of the Generator Configuration options can be used as Ecore Annotation. These options will be used by default; any external options take precedence over annotation options.

EcoreDoc will throw an `IllegalArgumentException` if an EcoreDoc annotation contains an illegal value.

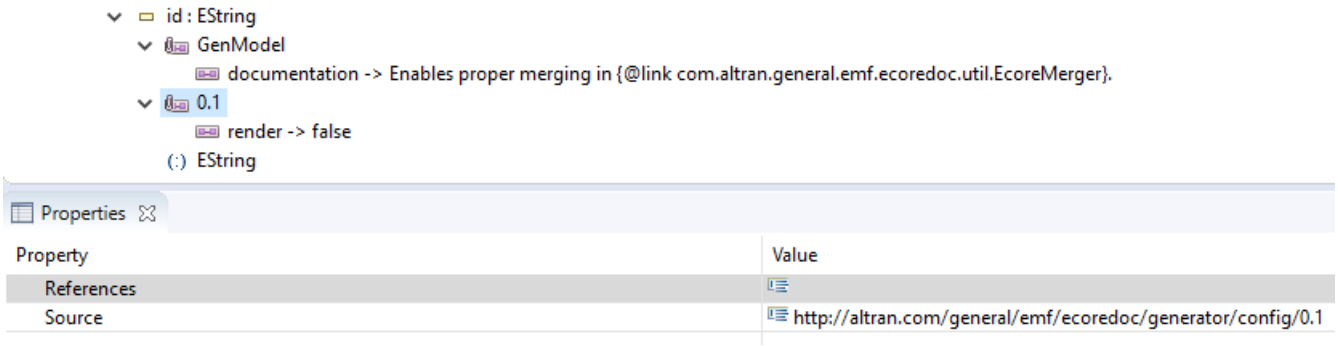
6.1. Ecore Annotation

Create an `EAnnotation` on the annotated element with source

```
http://altran.com/general/emf/ecoredoc/generator/config/0.1
```

Within this annotation, create one key/value pair for each option.

The source identifier might change in the future! However, it should be possible to maintain backwards compability.



Example Ecore annotation on EAttribute **id** *(sets render for* **id** *to* `false`*)*

6.2. Xcore Annotation

First, register the annotation.

EcoreDoc annotation registration

```
annotation "http://altran.com/general/emf/ecoredoc/generator/config/0.1" as EcoreDoc
```

Afterwards, we can use the annotation as usual.

Example Xcore annotation on EAttribute **name** *(sets render for* **name** *to* `false`*)*

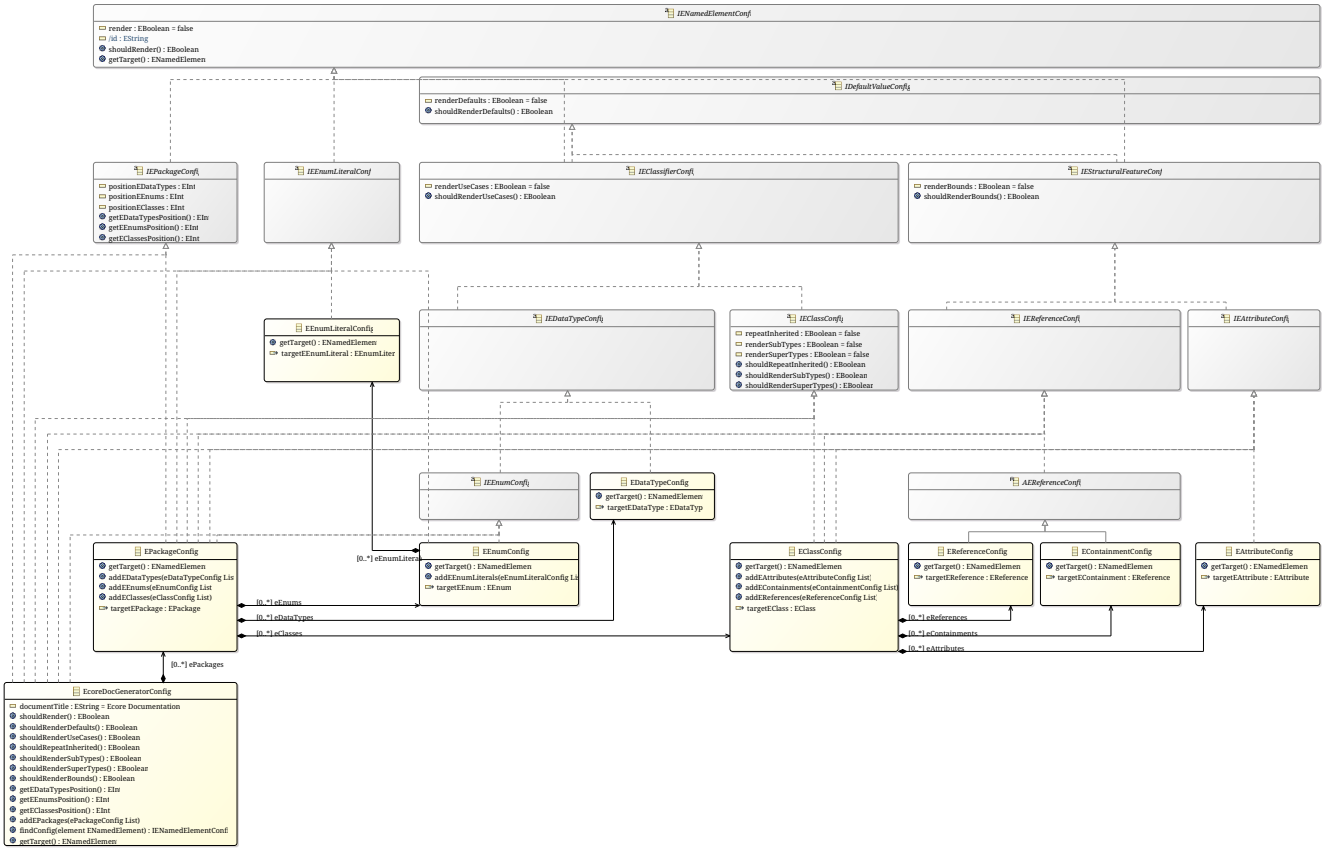
```
@EcoreDoc(
  render="false"
)
String name
```

7. Generator Configuration

The generator configuration is an Ecore metamodel, so we obviously use EcoreDoc to create the documentation listed below.

The most important parts are:

- Class EcoreDocGeneratorConfig as model root, also describing the customization hierarchy
- Interface IENamedElementConfig, implemented by all elements
- Interface IDefaultValueConfig, implemented by all elements except Class EEnumLiteralConfig
- Interface IEClassifierConfig
- Interface IEClassConfig
- Interface IEPackageConfig
- Interface IEStructuralFeatureConfig



Generator Configuration Class Diagram

7.1. Abstract Class AEReferenceConfig

Super-types

- `config.IDefaultValueConfig`
- `config.IENamedElementConfig`
- `config.IEReferenceConfig`
- `config.IEStructuralFeatureConfig`

7.2. Class EAttributeConfig

Super-types

- `config.IDefaultValueConfig`
- `config.IEAttributeConfig`
- `config.IENamedElementConfig`
- `config.IEStructuralFeatureConfig`

References

Name	Type	Properties	Description
`targetEAttribute`	`ecore.EAttribute`	`[0..1]`	

Used at

- `config.EClassConfig.eAttributes`

7.3. Class EClassConfig

Super-types

- config.IDefaultValueConfig
- config.IEAttributeConfig
- config.IEClassConfig
- config.IEClassifierConfig
- config.IENamedElementConfig
- config.IEReferenceConfig
- config.IEStructuralFeatureConfig

Containments

Name	Type	Properties	Description
eAttributes	config.EAttributeConfig	[0..*]	
eContainments	config.EContainmentConfig	[0..*]	
eReferences	config.EReferenceConfig	[0..*]	

References

Name	Type	Properties	Description
targetEClass	ecore.EClass	[0..1]	

Used at

- config.EPackageConfig.eClasses

7.4. Class EContainmentConfig

Super-types

- config.AEReferenceConfig
- config.IDefaultValueConfig
- config.IENamedElementConfig
- config.IEReferenceConfig
- config.IEStructuralFeatureConfig

References

Name	Type	Properties	Description
targetEContainment	ecore.EReference	[0..1]	

Used at

- config.EClassConfig.eContainments

7.5. Class EDataTypeConfig

Super-types

- config.IDefaultValueConfig
- config.IEClassifierConfig
- config.IEDataTypeConfig
- config.IENamedElementConfig

References

Name	Type	Properties	Description
targetEDataType	ecore.EDataType	[0..1]	

Used at

- config.EPackageConfig.eDataTypes

7.6. Class EEnumConfig

Super-types

- config.IDefaultValueConfig
- config.IEClassifierConfig
- config.IEDataTypeConfig
- config.IEEnumConfig
- config.IEEnumLiteralConfig
- config.IENamedElementConfig

Containments

Name	Type	Properties	Description
eEnumLiterals	config. EEnumLiteralConfig	[0..*]	

References

Name	Type	Properties	Description
targetEEnum	ecore.EEnum	[0..1]	

Used at

- config.EPackageConfig.eEnums

7.7. Class EEnumLiteralConfig

Super-types

- config.IEEnumLiteralConfig
- config.IENamedElementConfig

References

Name	Type	Properties	Description
targetEEnumLiteral	ecore.EEnumLiteral	[0..1]	

Used at

- config.EEnumConfig.eEnumLiterals

7.8. Class EPackageConfig

Super-types

- config.IDefaultValueConfig
- config.IEAttributeConfig
- config.IEClassConfig
- config.IEClassifierConfig
- config.IEDataTypeConfig
- config.IEEnumConfig
- config.IEEnumLiteralConfig
- config.IENamedElementConfig
- config.IEPackageConfig
- config.IEReferenceConfig
- config.IEStructuralFeatureConfig

Containments

Name	Type	Properties	Description
eClasses	config.EClassConfig	[0..*]	
eDataTypes	config.EDataTypeConfig	[0..*]	
eEnums	config.EEnumConfig	[0..*]	

References

Name	Type	Properties	Description
targetEPackage	ecore.EPackage	[0..1]	

Used at

- config.EcoreDocGeneratorConfig.ePackages

7.9. Class EReferenceConfig

Super-types

- config.AEReferenceConfig
- config.IDefaultValueConfig
- config.IENamedElementConfig
- config.IEReferenceConfig
- config.IEStructuralFeatureConfig

References

Name	Type	Properties	Description
targetEReference	ecore.EReference	[0..1]	

Used at

- config.EClassConfig.eReferences

7.10. Class EcoreDocGeneratorConfig

Root for the detailed EcoreDocGenerator configuration.

The configuration allows to specify configuration options for each element and all its contained elements. It always chooses the most specific configuration setting.

Example:

EcoreDocGeneratorConfig * renderDefaults: {unset, defaults to true} * repeatInherited: false + EPackage1 * renderDefaults: false + EClass1 + EAttribute1 * renderDefaults: true + EAttribute2 {no custom config} + EClass2 extends EClass1 + EPackage2 * repeatInherited: true + EClass3 extends EClass1 + EClass4 + EAttribute3 * renderDefaults: true * repeatInherited: false

Result:

```
EPackage1
```

```
renderDefaults false
repeatInherited false
```

```
EClass1
```

```
renderDefaults false
repeatInherited false
```

```
EAttribute1
```

```
renderDefaults true
repeatInherited false
```

```
EAttribute2
```

```
renderDefaults false
repeatInherited false
```

```
EClass2
```

```
renderDefaults false
repeatInherited false
```

```
EPackage2
```

```
renderDefaults true
repeatInherited true
```

```
EClass3
```

```
renderDefaults true
repeatInherited true
```

```
EClass4
```

```
renderDefaults true
repeatInherited true
```

```
EAttribute3
```

```
renderDefaults true
repeatInherited false
```

Super-types

- config.IDefaultValueConfig
- config.IEAttributeConfig
- config.IEClassConfig
- config.IEClassifierConfig
- config.IEDataTypeConfig
- config.IEEnumConfig

- config.IEEnumLiteralConfig
- config.IENamedElementConfig
- config.IEPackageConfig
- config.IEReferenceConfig
- config.IEStructuralFeatureConfig

Attributes

Name	Type	Properties	Description
documentTitle	EString	[0..1] *Default:* Ecore Documentation	Title of the generated document. defaults to Ecore Documentation.

Containments

Name	Type	Properties	Description
ePackages	config. EPackageConfig	[0..*]	

7.11. Interface IDefaultValueConfig

Attributes

Name	Type	Properties	Description
renderDefaults	EBoolean	[0..1] unsettable	Whether properties should be rendered at their default values. Example: If EReference.ordered = true (the default value), the ordered property of that EReference will not be rendered if renderDefaults = false.

7.12. Interface IEAttributeConfig

Super-types

- config.IDefaultValueConfig
- config.IENamedElementConfig
- config.IEStructuralFeatureConfig

7.13. Interface IEClassConfig

Super-types

- config.IDefaultValueConfig
- config.IEClassifierConfig
- config.IENamedElementConfig

Attributes

Name	Type	Properties	Description
renderSubTypes	EBoolean	[0..1] unsettable	Whether the list of sub-types should be rendered.
renderSuperTypes	EBoolean	[0..1] unsettable	Whether the list of super-types should be rendered.
repeatInherited	EBoolean	[0..1] unsettable	Whether inherited features should be repeated. Example: EClass1 has an EAttribute name=attr1. EClass2 extends EClass1. If repeatInherited = true for EClass2, attr1 will be listed in the section of EClass1 and EClass2. Otherwise, attr1 will only be listed in the section of EClass1.

7.14. Interface IEClassifierConfig

Super-types

- config.IDefaultValueConfig
- config.IENamedElementConfig

Attributes

Name	Type	Properties	Description
renderUseCases	EBoolean	[0..1] unsettable	Whether use cases (references to other usages of this element) should be rendered.

7.15. Interface IEDataTypeConfig

Super-types

- config.IDefaultValueConfig
- config.IEClassifierConfig
- config.IENamedElementConfig

7.16. Interface IEEnumConfig

Super-types

- config.IDefaultValueConfig
- config.IEClassifierConfig
- config.IEDataTypeConfig

- config.IENamedElementConfig

7.17. Interface IEEnumLiteralConfig

Super-types

- config.IENamedElementConfig

7.18. Interface IENamedElementConfig

Attributes

Name	Type	Properties	Description
render	EBoolean	[0..1] unsettable	Whether this element should be rendered at all.

7.19. Interface IEPackageConfig

Super-types

- config.IENamedElementConfig

Attributes

Name	Type	Properties	Description
positionEClasses	EInt	[0..1] unsettable	Rendering position of all EClasses within an EPackage.
positionEDataTypes	EInt	[0..1] unsettable	Rendering position of all EDataTypes within an EPackage.
positionEEnums	EInt	[0..1] unsettable	Rendering position of all EEnums within an EPackage.

7.20. Interface IEReferenceConfig

Super-types

- config.IDefaultValueConfig
- config.IENamedElementConfig
- config.IEStructuralFeatureConfig

7.21. Interface IEStructuralFeatureConfig

Super-types

- config.IDefaultValueConfig
- config.IENamedElementConfig

Attributes

Name	Type	Properties	Description
`renderBounds`	`EBoolean`	`[0..1]` unsettable	Whether multiplicity bounds should be rendered, even if they are at their default values and renderDefaults = false.

8. Versions

This asset in version 0.8.0-SNAPSHOT was developed using the following components and versions.

Eclipse

4.7.3a (Oxygen 3a)

Google Guava

19.0

Apache Commons Lang3

3.4

Apache Commons IO

2.2

Apache Maven

3.3.9

Eclipse Ecore

2.12.0

Eclipse Xcore

1.3.1

Eclipse Tycho

1.2.0

9. Known Issues

- If HTML is used in Ecore documentation, the PDF rendering can be faulty (Issue #12)
- `EOperation`s, `EParameter`s, `EAnnotation`s are missing from the documentation (Issue #13, Issue #15)