NAME

yauml - A script for generating UML diagrams from YAML file.

SYNOPSYS

yauml [-t *template_file***] [-o** *out_file* **[-T***type***]]** *file.yaml*

yauml [-hv]

DESCRIPTION

This program generates a formatted string from an YAML file according to the dot programming language interpreted by **Graphviz**. The output can then be used by **dot** in order to get a file in png, pdf or another file format supported (see **dot**(1)). However, **yauml** handles **-T** and **-o** options just like **dot**. If those options are given, the program will automatically call **dot** as a sub-process.

OPTIONS

-t, --template *template_file*
: The template file to use (default: **TEMPLATE_FILE**).

-o, --out *out_file*
: The file to output the generated result (default: **stdout**).

-T, --Type *format*
: The type of file to generate (see **dot**(1)). Specifying this will automatically pass the output of **yauml** to **dot**. **This option has to be used with -o**.

-h, --help
: Shows a help text on **stdout**.

-v, --version
: Version of the program.

YAML format

The program will scan the YAML file for *class* and *interface* blocks written like so:

```
- class: ClassName
  # rest of block

- interface: InterfaceName
  # rest of block
```

Class block

A class block can contain *attributes*, *methods*, *implements*, *ispartof* and *inherits* sub-block. For e.g:

```
- class: A
  attributes:
  - att1
  - att2
  - ...
  methods:
  - method1()
  - method2()
  - ...
```

Considering ths class **A**, the following could be added:

```
- class: B
  inherits:
  - A
  ...
```

This would mean that **B** inherits from **A**. *implements* and *ispartof* blocks would be used exactly the same way as *inherits*. In order to specify the **multiplicity** of the entities involved in the relation:

```
- class: B
inherits:
- A
ispartof:
- C [1..*] [1]
```

where the first and second string between brackets respectively mean the multiplicity of the child and the parent.

Abstract classes are written like so:

```
- class: C { Abstract }
  methods:
  - method1() { Abstract }
  - method2()
  - ...
```

Interface block

Same as a class block but only *methods* sub-blocks are treated.

FILES

TEMPLATE_FILE

/usr/share/yauml/template.dot

This file is written accordingly to the **dot** programming language. However, in order to properly use **yauml**, it is required to place some **flags** in the file so that the program finds where to inject the right string at the right place. Those flags are the following:

```
// CLASSES

// INTERFACES

// SIMPLE RELATIONS

// USE RELATIONS

// INHERIT RELATIONS

// ISPARTOF RELATIONS

// IMPLEMENT RELATIONS
```

All of these flags consist in a commented line containing the word "CLASSES" or "INTERFACES" or ... in capital letters (note that "// The CLASSES here" would be valid too). The first occurence of each flags

will be followed by the respective string built from the YAML file.

AUTHORS

Written by

- Alexandre Blondin Massé <alexandre.blondin.masse@gmail.com>
- Simon Désaulniers <rostydela@gmail.com>

SEE ALSO

dot(1)