coqdoc: a documentation tool for Coq

Jean-Christophe Filliâtre
http://www.lri.fr/~filliatr/coqdoc

Contents

1 Introduction

coqdoc is a documentation tool for the proof assistant Coq, similar to javadoc or ocamldoc. The task of coqdoc is

1. to produce a nice LaTeX and/or HTML document from the Coq sources, readable for a human and not only for the proof assistant;
2. to help the user navigating in his own (or third-party) sources.

2 Principles

Documentation is inserted into Coq files as *special comments*. Thus your files will compile as usual, whether you use coqdoc or not. coqdoc presupposes that the given Coq files are well-formed (at least lexically). Documentation starts with `(**`, followed by a space, and ends with the pending `*)`. The documentation format is inspired by Todd A. Coram's *Almost Free Text (AFT)* tool: it is mainly ASCII text with some syntax-light controls, described below. coqdoc is robust: it shouldn't fail, whatever the input is. But remember: "garbage in, garbage out".

Coq material inside documentation. Coq material is quoted between the delimiters `[` and `]`. Square brackets may be nested, the inner ones being understood as being part of the quoted code (thus you can quote a term like $[x : T]u$ by writing `[[x:T]u]`). Inside quotations, the code is pretty-printed in the same way as it is in code parts.

Pre-formatted vernacular is enclosed by `[[` and `]]`. The former must be followed by a newline and the latter must follow a newline.

Pretty-printing. `coqdoc` uses different faces for identifiers and keywords. The pretty-printing of Coq tokens (identifiers or symbols) can be controlled using one of the following commands:

```
(** printing token %...LaTeX...% #...HTML...# *)
```

or

```
(** printing token $...LaTeX math...$ #...HTML...# *)
```

It gives the LaTeX and HTML texts to be produced for the given Coq token. One of the LaTeX or HTML text may be ommitted, causing the default pretty-printing to be used for this token.

The printing for one token can be removed with

```
(** remove printing token *)
```

Initially, the pretty-printing table contains the following mapping:

`->`	$\rightarrow$	`<-`	$\leftarrow$	`*`	$\times$
`<=`	$\leq$	`>=`	$\geq$	`=>`	$\Rightarrow$
`<>`	$\neq$	`<->`	$\leftrightarrow$	`|-`	$\vdash$
`\/`	$\vee$	`/\`	$\wedge$	`~`	$\neg$

Any of these can be overwritten or suppressed using the `printing` commands.

Important note: the recognition of tokens is done by a (ocaml)lex automaton and thus applies the longest-match rule. For instance, `->~` is recognized as a single token, where Coq sees two tokens. It is the responsability of the user to insert space between tokens *or* to give pretty-printing rules for the possible combinations, e.g.

```
(** printing ->~ %\ensuremath{\rightarrow\lnot}% *)
```

Sections. Sections are introduced by 1 to 4 leading stars (i.e. at the beginning of the line). One star is a section, two stars a sub-section, etc. The section title is given on the remaining of the line. Example:

```
    (** * Well-founded relations

        In this section, we introduce...  *)
```

Lists. List items are introduced by 1 to 4 leading dashes. Deepness of the list is indicated by the number of dashes. List ends with a blank line. Example:

```
This module defines
    - the predecessor [pred]
    - the addition [plus]
    - order relations:
      -- less or equal [le]
      -- less [lt]
```

Rules. More than 4 leading dashes produce an horizontal rule.

Escapings to LaTeX and HTML. Pure LaTeX or HTML material can be inserted using the following escape sequences:

- `$...LaTeX stuff...$` inserts some LaTeX material in math mode. Simply discarded in HTML output.
- `%...LaTeX stuff...%` inserts some LaTeX material. Simply discarded in HTML output.
- `#...HTML stuff...#` inserts some HTML material. Simply discarded in LaTeX output.

Verbatim. Verbatim material is introduced by a leading `<<` and closed by `>>`. Example:

```
Here is the corresponding caml code:
<<
  let rec fact n =
    if n <= 1 then 1 else n * fact (n-1)
>>
```

Hyperlinks. Hyperlinks can be inserted into the HTML output, so that any identifier is linked to the place of its definition.

In order to get hyperlinks you need to first compile your Coq file using `coqc --dump-glob` *`file`*; this appends Coq names resolutions done during the compilation to file *`file`*. Take care of erasing this file, if any, when starting the whole compilation process.

Then invoke `coqdoc --glob-from` *`file`* to tell `coqdoc` to look for name resolutions into the file *`file`*.

Identifiers from the Coq standard library are linked to the Coq web site at `http://coq.inria.fr/library/`. This behavior can be changed using command line options `--no-externals` and `--coqlib`; see below.

Hiding / Showing parts of the source. Some parts of the source can be hidden using command line options `-g` and `-l` (see below), or using such comments:

```
(* begin hide *)
some Coq material
(* end hide *)
```

Conversely, some parts of the source which would be hidden can be shown using such comments:

```
(* begin show *)
some Coq material
(* end show *)
```

The latter cannot be used around some inner parts of a proof, but can be used around a whole proof.

3 Usage

coqdoc is invoked on a shell command line as follows:

coqdoc < *options and files* >

Any command line argument which is not an option is considered to be a file (even if it starts with a -). Coq files are identified by the suffixes `.v` and `.g` and LaTeX files by the suffix `.tex`.

HTML output

This is the default output. One HTML file is created for each Coq file given on the command line, together with a file `index.html` (unless option `-no-index` is passed). The HTML pages use a style sheet named `style.css`. Such a file is distributed with coqdoc.

LaTeX output

A single LaTeX file is created, on standard output. It can be redirected to a file with option `-o`. The order of files on the command line is kept in the final document. LaTeX files given on the command line are copied 'as is' in the final document . DVI and PostScript can be produced directly with the options `-dvi` and `-ps` respectively.

TeXmacs output

To translate the input files to TeXmacs format, to be used by the TeXmacs Coq interface (see `http://www-sop.inria.fr/lemme/Philippe.Audebaud/tmcoq/`).

Command line options

Overall options

`--html`

Select a HTML output.

`--latex`

Select a LaTeX output.

`--dvi`

Select a DVI output.

`--ps`

Select a PostScript output.

`--texmacs`

Select a TeXmacs output.

`-o` ***file***, `--output` ***file***

Redirect the output into the file '*file*' (meaningless with `-html`).

`-d` ***dir***, `--directory` ***dir***

Output files into directory '*dir*' instead of current directory (option `-d` does not change the filename specified with option `-o`, if any).

`-s` , `--short`

Do not insert titles for the files. The default behavior is to insert a title like "Library Foo" for each file.

`-t` ***string***, `--title` ***string***

Set the document title.

`--body-only`

Suppress the header and trailer of the final document. Thus, you can insert the resulting document into a larger one.

`-p` ***string***, `--preamble` ***string***

Insert some material in the LaTeX preamble, right before `\begin{document}` (meaningless with `-html`).

`--vernac-file` ***file***, `--tex-file` ***file***

Considers the file '*file*' respectively as a `.v` (or `.g`) file or a `.tex` file.

`--files-from` ***file***

Read file names to process in file '*file*' as if they were given on the command line. Useful for program sources splitted in several directories.

`-q`, `--quiet`

Be quiet. Do not print anything except errors.

`-h`, `--help`

Give a short summary of the options and exit.

`-v`, `--version`

Print the version and exit.

Index options

Default behavior is to build an index, for the HTML output only, into `index.html`.

`--no-index`

Do not output the index.

`--multi-index`

Generate one page for each category and each letter in the index, together with a top page `index.html`.

Table of contents option

`-toc`, `--table-of-contents`

Insert a table of contents. For a LaTeX output, it inserts a `\tableofcontents` at the beginning of the document. For a HTML output, it builds a table of contents into `toc.html`.

Hyperlinks options

`--glob-from` ***file***

Make references using Coq globalizations from file *file*. (Such globalizations are obtained with Coq option `-dump-glob`).

`--no-externals`

Do not insert links to the Coq standard library.

`--coqlib` ***url***

Set base URL for the Coq standard library (default is `http://coq.inria.fr/library/`).

`-R` ***dir coqdir***

Map physical directory *dir* to Coq logical directory *coqdir* (similarly to Coq option `-R`).

Note: option `-R` only has effect on the files *following* it on the command line, so you will probably need to put this option first.

Contents options

`-g`, `--gallina`

Do not print proofs.

`-l`, `--light`

Light mode. Suppress proofs (as with `-g`) and the following commands:

- [`Recursive`] `Tactic Definition`

- `Hint / Hints`
- `Require`
- `Transparent / Opaque`
- `Implicit Argument / Implicits`
- `Section / Variable / Hypothesis / End`

The behavior of options `-g` and `-l` can be locally overridden using the `(* begin show *)` ... `(* end show *)` environment (see above).

Language options
Default behavior is to assume ASCII 7 bits input files.

`-latin1`, `--latin1`

Select ISO-8859-1 input files. It is equivalent to `-inputenc latin1 -charset iso-8859-1`.

`-utf8`, `--utf8`

Select UTF-8 (Unicode) input files. It is equivalent to `-inputenc utf8 -charset utf-8`. LaTeX UTF-8 support can be found at `http://www.ctan.org/tex-archive/macros/latex/contrib/supported/unicode/`.

`--inputenc` ***string***

Give a LaTeX input encoding, as an option to LaTeX package `inputenc`.

`--charset` ***string***

Specify the HTML character set, to be inserted in the HTML header.

4 The coqdoc LaTeX style file

In case you choose to produce a document without the default LaTeX preamble (by using option `--no-preamble`), then you must insert into your own preamble the command

```
\usepackage{coqdoc}
```

Then you may alter the rendering of the document by redefining some macros:

`coqdockw`, `coqdocid`

The one-argument macros for typesetting keywords and identifiers. Defaults are sans-serif for keywords and italic for identifiers.

For example, if you would like a slanted font for keywords, you may insert

```
\renewcommand{\coqdockw}[1]{\textsl{#1}}
```

anywhere between `\usepackage{coqdoc}` and `\begin{document}`.

`coqdocmodule`

One-argument macro for typesetting the title of a `.v` file. Default is

```
\newcommand{\coqdocmodule}[1]{\section*{Module #1}}
```

and you may redefine it using `\renewcommand`.