DOC++ 3.4.9 Reference Card

by Steven R. Gould

Comment Styles

Every DOC++ comment defines a manual entry. A manual entry consists of documentation from the DOC++ comment and information from the subsequent declaration.
Trailing comments can be used to define manual entries. Enable these by turning on Quantel extensions (-Q).

/** ... */	C style
/// ...	C++ style
//@{ ... //@}	Defines documentation scope/grouping
//@Include:*file*	Includes named file. Wildcards allowed.

DOC++ Declaration Tags

These tags are used immediately before a subsequent declaration. Note that in the following section, the term function refers to a method of a class, or a standalone function.

Tag	Description
@author	Author of function
@deprecated	Function has been deprecated
@doc	Long documentation
@exception	Documents exceptions thrown
@field	Documents fields
@invariant	Documents invariants
@memo	Short documentation (typically, one line)
@name	Function name (overrides DOC++)
@param	Documents the named parameter
@postcondition	Documents postconditions
@precondition	Documents preconditions
@return	Documents functions return value
@see	Cross reference.
@since	Version when function was introduced
@version	Current version

Inline DOC++ Tags

The following tags are useful when writing documentation for a function, method, variable, and so on.

Tag	Description
#text#	Corresponds to TeX "\verb!*foo*!" - outputs "*foo*" verbatim.
@filename *file*	Force manual entry to go to the named file
{@link *entry name*}	Cross-reference to manual entry. *name* specifies link name (optional).
\Date	Insert current date and time
\IMG{*file*}	Insert image into documentation
\IMG[*param*]{*file*}	Insert image, using HTML parameters
\IMG[*param*][*param*]{*file*}	Insert image, using HTML and TeX params, resp. For TeX output, include the 'graphicx' TeX package
\Label{*label*}	Make a label
\Ref{*entry*}	Cross-reference to manual entry
\URL{*URL*}	Make link to web page, *URL*
\URL[*name*]{*URL*}	Make link to web page, with *text* as the link text
\TEX{*text*}	Include the TeX "*text*" in document - for HTML output, generates GIFs
\includegraphics	Same as \IMG
\today	Same as \Date

Supported HTML macros

For best results with both printed (TeX) and online (HTML) documentation, it is recommended that TeX macros are used - see next section - and not the HTML macros given below.

Tag	Description
 	New line; i.e. line break
<P>	New paragraph
<EM>*text*</EM>	Emphasize *text* - usually means italic
<I>*text*</I>	Italicize *text*
<B>*text*</B>	Make *text* bold
<STRONG>*text*</STRONG>	Make *text*"strong" - usually bold
<TT>*text*</TT>	Display verbatim - don't format
<PRE>*text*</PRE>	Display verbatim - don't format
<CODE>*text*</CODE>	Display verbatim - don't format
<OL>...</OL>	Ordered/numbered list
<DL>...</DL> <DT>...</DT> <DD>...</DD>	Description
<UL>...</UL>	Unordered/bulleted list
<LL>...</LL>	Enumerations
<LI>	List item, used within <OL> or <UL>

Supported TeX macros

For best results with both printed (TeX) and online (HTML) documentation, it is recommended that TeX macros are used

Tag	Description
$...$	math mode for inline equations
\[...\]	display math mode
\#	Output character "#"
_	Output character "_"
\	Output character " " (a space)
\hline	Horizontal line

The following macros are all of the form:
{*name text*}
The macro applies to all *text* enclosed within the curly brackets, {}, and following the macro *name*.

Macro	Description
\em	Emphasize enclosed characters
\bf	Bold face for enclosed characters
\it	Italicize enclosed characters
\tt	Use fixed font for enclosed characters
\tiny	Use small font for enclosed characters
\scriptsize	Use script size for enclosed characters
\footnotesize	Use footnote size for enclosed characters
\small	Use small font for enclosed characters
\large	Use large font for enclosed characters
\Large	Use Large font for enclosed characters
\LARGE	Use LARGE font for enclosed characters
\huge	Use huge font for enclosed characters
\Huge	Use Huge font for enclosed characters
\HUGE	Use HUGE font for enclosed characters

The following TeX macros are all of the form:
\begin{*name*} ... \end{*name*}
For brevity only the *name*of the tag is given below.

Macro	Description
center	Center paragraph
flushleft	Left align paragraph
flushright	Right align paragraph
verbatim	Output enclosed text as is
tabular	Defines a table
array	Defines an array
itemize	Defines a bulleted list of items
enumerate	Defines a numerated list of items
description	Description (???)
equation	Defines an equation
eqnarray	Equation array (???)

DOC++ 3.4.9 Reference Card

Command Line Options

Input Formats

If none of the following input options are used, DOC++ defaults to C++/C mode.

Short	Long	Source file format
		C++/C mode (default)
`-J`	`--java`	Java files
`-Y`	`--idl`	IDL files
`-z`	`--php`	PHP files

Output Formats

If none of the following output options are used, DOC++ defaults to output in HTML format.

Short	Long	Output file format
		HTML (default)
`-t`	`--tex`	TeX (PS/PDF)
`-Z`	`--docbook`	DocBook (SGML)

General Options

Short	Long	Description
`-A`	`--all`	Generate entry for all elements
`-c`	`--c-comments`	C/C++ comments as DOC++ comments
`-C` *file*	`--config` *file*	Read options
`-h`	`--help`	Display available options
`-H`	`--html`	Use HTML as formatting language
`-I` *file*	`--input` *file*	Read list of input files from *file*
`-J`	`--java`	Parse Java source files
`-nd`	`--no-define`	Ignore `#define` macros
`-ng`	`--no-class-graph`	Don't generate class graphs
`-p`	`--private`	Include private members in output
`-q`	`--quick`	Run fast, even if larger file output
`-Q`	`--quantel`	Parse Quantel extensions
`-R`	`--internal-doc`	Generate internal documentation
`-t`	`--tex`	Produce TeX output
`-u`	`--upwards-arrow`	Draw upwards arrows in class graphs
`-v`	`--verbose`	Put DOC++ in verbose mode
`-V`	`--version`	Output DOC++ version info.
`-y`	`--scan-includes`	Scan `#include`'ed header files
`-Y`	`--idl`	Parse IDL source files
`-z`	`--php`	Parse PHP source files
`-Z`	`--docbook`	Output DocBook SGML

Customizing HTML Output

HTML-specific Command Line Options

These options are only active when HTML output is selected, i.e. when no `-t` or `--tex` option is used.

Short	Long	Description
`-a`	`--tables`	Use HTML tables
`-b`	`--tables-border`	Use HTML tables, with borders
`-B` *file*	`--footer` *file*	Use *file* as HTML footer
`-d` *name*	`--dir` *name*	Specifies output directory for HTML
`-f`	`--filenames`	Output source file name on each page
`-F`	`--filenames-path`	As above, but output full path
`-g`	`--no-gifs`	Don't generate GIFs for equations/TeX
`-G`	`--gifs`	Force re-generation of GIFs
`-i`	`--no-inherited`	Don't show inherited members
`-j`	`--no-java-graphs`	Don't use Java applets for class graphs
`-k`	`--trivial-graphs`	Generate even trivial class graphs
`-K` *file*	`--stylesheet` *file*	Use *file* as stylesheet for generated pages
`-m`	`--no-members`	Hide members with no documentation
`-M`	`--full-toc`	Show members in Table of Contents
`-P`	`--no-general`	Discard general stuff
`-S`	`--sort`	Sort entries alphabetically
`-T` *file*	`--header` *file*	Use *file* as the HTML header
`-w`	`--before-group`	Print groups' doc. before groups
`-W`	`--before-class`	Print classes' doc. before classes
`-x` x	`--suffix` x	Use x as file extension, instead of `.html`

Customizing HTML Pages

In addition to the above command line options, the HTML output can be customized by defining any combination of the following files. They will be inserted in the appropriate places on the relevant pages.

File name	Description
`indexHeader.inc`	Header for index pages
`indexFooter.inc`	Footer for index pages
`hierHeader.inc`	Header for class hierarchy pages
`hierFooter.inc`	Footer for class hierarchy pages
`classHeader.inc`	Header for all other pages
`classFooter.inc`	Footer for all other pages

The `indexHeader.inc` and `hierHeader.inc` files should start with `<HTML><TITLE>...<BODY>`, whereas `classHeader.inc` should start with `<BODY>` (since DOC++ sets up the title).

Within these files, certain special tags are supported, as listed below. DOC++ will substitute these when generating the HTML pages.

Tag	Description
`%file`	Entry's file name
`%fullname`	Entry's full name (includes the inheritance)
`%name`	Entry's name
`%type`	Entry's return type

Customizing TeX Output

TeX-specific Command Line Options

These options provide control over the TeX output of DOC++.

Short	Long	Description
`-ec`	`--class-graph`	Only generates the class graph
`-ef` *file*	`--env` *file*	Read TeX environment from *file*
`-ei`	`--index`	Only generate the index
`-eo` *option*	`--style` *option*	Adds *option* to TeX's `\documentclass`
`-ep` *name*	`--package` *name*	Adds `\usepackage{`*name*`}` to TeX env.
`-et` *file*	`--title` *file*	Use *file* as TeX title page
`-D` x	`--depth` x	Sets min. depth in Table of Contents
`-l`	`--no-env`	Disable generation of TeX environmen
`-o` *file*	`--output` *file*	Sets the output filename
`-s`	`--source`	Generate formatted source code listing
`-X`	`--hide-index`	Disable index at start of each section

Customizing the TeX Document

In addition to the above command line options, the TeX output can be customized by editing the style file `docxx.sty` (Sorry, but there is no documentation on how to do this.)

Example Command Lines

To generate HTML documentation, I often use something like:

```
doc++ -p -u -d docs/html -B docs/banner.html \
                              docs/name.dxx
```

This includes private members, uses up-arrows in class graphs, stores the resulting HTML files in the docs/html subdirectory, uses the docs/banner.html file as a footer on each page, and reads docs/*name*.dxx as the main DOC++ input file.

Similarly, to generate TeX documentation (which is then processed to create a PDF or PS file), I use:

```
doc++ -p -u -t -o docs/latex/name.tex docs/name.dxx
```

Reference

DOC++ project	`http://sourceforge.net/projects/docpp/`
DOC++ home page	`http://docpp.sourceforge.net/`
Author's web site	`http://www.stevengould.org/`

Send comments and corrections to Steven R. Gould, Dallas, TX 75252, USA. (steven.gould@stevengould.org)