Doxygen Quick Reference[1]

© Csaba Szepesvari, 2003
www.sztaki.hu/~szcsaba
v1.0

Notational syntax

Syntax	Scope, meaning
<arg>	Single word
(arg)	End of the line on which the command was found
{arg}	Next paragraph[2]
[arg]	Optional argument

Comment blocks – use ///, ///<, turn on JAVADOC_AUTOBRIEF, put brief docs at declarations, detailed docs at definitions.[3]

Structural organization - It is wise to have separate documentation files for your projects (myproject.h), namespaces (mynamespace.h), modules (mymodule.h) and other pages (mypage.h).

Mainpage, page – Purpose is to have a main index + documentation. Typical mainpage:

```
\section Introduction
ONE PARAGRAPH DESCRIPTION OF THE
PACKAGE/PROJECT
\section Overview
ORGANIZATION PRINCIPLES, PARTS
\subsection Part1
..
\section Additional Resources
RECOMMENDED PRACTICES USING PROJECT,
TEST CODE, CODING GUIDELINES USED,
TODO PAGES, EXTERNAL LINKS.
```

\mainpage [(title)]	Main page documentation[4]
\page <name> (title)	Separate documentation page
\section <name> (title) \subsection, \subsubsection,	Creates a section named <name> and title <title>. \subsection, \subsubsection, \par
\par	work in the same way.
\anchor <word>	Invisible, named anchor; can be used with \ref.[5]

Groups –Purpose is to provide a means for documenting semantically related objects together. Groups exist in two flavours: *modules* & *member groups*. Modules get a separate page. Members of modules can be files, namespaces, classes, functions, variables, enums, typedefs, and defines, but also other groups. Commands to create and organize modules are \addtogroup, \defgroup, \ingroup, and \weakgroup.
Member groups group together things (typically class members) that are also physically grouped in the source file. No nesting is allowed here. Use \\{@ and \\}@ to enclose group members – this can be used to group both module and member groups members.

\addtogroup <name> [(title)]	Incremental group definition
\defgroup <name> (title)	Defines group with given name and title.
\ingroup (<gr1> [<gr2> <gr3 >])	Links block into a group or groups; applies to comment block of a class, file or namespace
\name (header)	Turns a comment block into a header definition of a member group. **Must** be followed by a \\{@ .. \\}@ block.
\weakgroup <name> [(title)]	Similar to \addtogroup, but has a lower priority when it comes to resolving conflicting grouping definitions.
\\@{,..\\@}	Starts and closes a group.

Out-of-order Documentation

\class <name> [<headerfile>] [<headername>]
\def <name>[6]
\enum <name>
\file [<name>]
\mainpage [(title)]
\namespace <name>
\page <name> (title)
\struct <name> [<headerfile>] [<headername>]
\typedef (typedef declaration)
\union <name> [<headerfile>] [<headername>]
\var (variable declaration)

[1] This document reflects my special needs and views. In particular it is not meant to be a complete reference guide to Doxygen commands.
[2] Paragraphs are delimited by a blank line or by a section indicator.
[3] By the implementation use \\\.<NL>\\\ DOC
[4] Use \section and friends to provide a structure of the main page
[5] Works only on \page and \mainpage.
[6] Macro definition

Documentation Sectioning

\author {author list}
\bug {description}
\date {description}
\deprecated {description}
\exception <object> {description}
\invariant {description}
\note {text}
\param <name> {description}[7]
\post {description}
\pre {description}
\remarks {text }
\return { description}
\retval <value> {description}
\todo {description}
\version {description}
\warning {description}

Cross References, Links

URLs and mail addresses found in the documentation are automatically replaced by links. All words in the documentation that correspond to a documented objects will automatically be replaced by a link. Suppress links with %.

\sa {list}	cross-references to classes, functions, methods, variables, files or URL; NAME::NAME refers to class member, just like NAME#NAME. Argument types can be used to select among overloaded function.
\link <link-object> TEXT \endlink	TEXT will be hyperlinked to <link-object>
\ref <name> ["(text)"]	References a named section, subsection, page or anchor.

Visual Enhancement

\a <word>	Arguments; same as \p
\b <word>	Boldface
\c <word>	Typewrite font (for code)
\code BLOCK \endcode	Block of code
\e <word>	Emphasizes word
\f$ FORMULA \f$	Inline formula
\f[FORMULA \f]	Displayed formula
\n	New line; same as \br
\par [(title)] {text}	Starts new paragraph – indented

[7] You can also document parameters by putting ///< after the parameter (if you list one parameter per line).

Special Commands

\copydoc <link-object>	Copies to doc of the referenced block

Graphs, Images

\dot DOT-GRAPH \enddot	Produces a dot-graph, can be made clickable.
\dotfile <file> ["caption"]	Insert a dotfile generated by dot. Search path is given by the DOT_PATH tag.
\image <format> <file> ["caption"] [<sizeindication>=<size>]	Inserts an image into the documentation. Search path is given by the IMAGE_PATH tag.

Lists

Use '-' to precede list items, aligned on the same column. Use '-#' to create a numbered list. Lists can be nested. Use '.' to indicate that a list is closed.

```
/// Text before the list
/// - list item 1
///  - sub item 1
///    - sub sub item 1
///    - sub sub item 2
///    .
///    The dot ends the sub sub item
///    list.
///    More text for the first sub item
///  .
///  The dot above ends the first sub
///  item.
///  More text for the first list item
///  - sub item 2
///  - sub item 3
/// - list item 2
/// .
/// More text in the same paragraph.
///
/// More text in a new paragraph.
///
```

Tagfiles – Purpose is to modularize documentation. A tag-file created for a project can be used in another project to generate links to the documenation of the source project.
To generate a tag-file specify the name of the tag file after GENERATE_TAGFILE in the config file of the project. To use the tag-file generated, list it in the TAGFILES list of the project that should use it. Config time link generation: put TAG_FILE_NAME=HTMLDOC_LOC. Use relative paths! (and slash not backslash).