Doxygen Quick Reference

Doxygen commands

Doxygen (http://www.doxygen.org/) reads a configuration file to control source code processing and documentation output formats, the default filename is `Doxyfile`.

`doxygen`	Run with default config file.
`doxygen <config-file>`	Run with `<config-file>`.
`doxygen -g <config-file>`	Generate `<config-file>`.

Documenting the sources

Doxygen-visible multi-line comment blocks begin with `/**`, or `/*!`, and end with `*/`. Alternately, the of C++ single line comment delimiters `///` or `//!` may be used on each line. Within a comment block HTML tags or Doxygen specific markup tags, **Special Commands**, can be used. Documentation comment blocks can occur anywhere in the source code but placing such blocks before defined code elements, classes, functions, etc., is recommended. Source code files should include the `\file` command to make comments in the file visible to Doxygen.

Doxygen special commands, with a few exceptions, begin with the prefix `@` or \, used in this document. Following the Doxygen manual convention, the command arguments are enclosed here in braces that signify the extent of the argument, these braces are not part of the command, nor should they be included in the command:

`< angle >`	angle braces: argument is a single word.
`( round )`	round braces: argument extends to end of line.
`{ curly }`	curly braces: argument extends to next paragraph.

Lists

Column aligned hypens (`-`) create a unordered or bulleted list. Column aligned hypens with a pound-number symbol (`-#`) create an ordered or numbered list. Lists can be made heirachical using indentation levels, list items with identical column aligned symbols will appear at the same level of the heirarchy. Unordered and ordered lists can be mixed in a heirarchy.

`-`	list item in unorderd list (column aligned).
`-#`	list item in orderd list (column aligned).
`\li`	list item in unorderd list.
`\arg`	equivalent to `\li`.
`<ul>`	HTML: starts an unordered list.
`</ul>`	HTML: ends an unordered list.
`<ol>`	HTML: starts an ordered list.
`</ol>`	HTML: ends an ordered list.
`<li>`	HTML: list item, between `<[uo]l>` and `</[uo]l>` pairs.

Grouping

Doxygen can group things in many ways. Groups are defined with either a `\defgroup`, `\name` or `\page` command with a label and optional title. The label can be used in other parts of the documentation as the first argument to a `\ref <lbl> ``<lnk>''` command. This creates a link to the labeled group using the `lnk` text enclosed in quotes, the second argument is optional. The group members are enclosed by the group open `@{` and close `@}` commands, as follows:

```
/** \defgroup <label> ``<title>'' */
/* @{ */
   // group members here ...
/* @} */
```

where groups initiated by `\name` or `\page` commands have similar forms, see table below. The open and close group commands must be placed in a comment block, this can be a standard C-style comment or single line C++ comment. Elements may be members of multiple groups, which may lead to conflicts when generating documentation. Doxygen implements a priority scheme for group membership. The priorities are assigned based on the command used to initiate group membership. The priority order (highest to lowest) is `\ingroup`, `\defgroup`, `\addtogroup`, and `\weakgroup`

`\defgroup <l> ``<t>''`	defines a module group with a label `l` and title `t`.
`\name <l>`	defines a member group with a label `l`.
`\page <l> (t)`	defines a page group with a label `l` and title `t`.
`\section <l> (t)`	defines a section on a page with a label `l` and title `t`.
`\subsection <l> (t)`	defines a sub-section on a page with a label `l` and title `t`.
`\mainpage (t)`	defines a documentation block to place on the index page.
`\addtogroup <l> (t)`	adds to a group, enclose in open-close pairs.
`\ingroup <l>`	adds documentation to the `l` group.
`\ref <l> ``<t>''`	references group `l` using the optional link text `t`.
`/* @{ */`	comment to open a group block
`/* @} */`	comment to close a group block
`// @{`	comment to open a group block
`// @{`	comment to open a group block

Module Groups

Module group documentation will appear under the **Modules** heading in the generated documentation. Module groups are defined with the `\defgroup <label> ``<title>''`, which gives the group a `label` for later reference. and a `title` to display in the documenation; title includes everything enclosed by quotes.

Member Groups

Member group are used when "a compound (e.g. a class or file) has many members, it is often desired to group them together". Member group documentation will not appear under under a separate header-tab in the generated documentation. Module groups are defined with the `\name <label>` command in a comment block which is taken as the group header, which gives the group a `label` for later reference.

Page Groups

Page group documentation will appear under the **Related Pages** heading in the generated documentation. Page groups are defined with the `\page <label> (title)`, which gives the group a `label` for later reference. and a `title` to display in the documenation. Pages can be further divided into sub-pages—with `\sub-page <n> ``<t>''`, sections and sub-sections, and sub-subsections with a command similar to: `\section <section-name> (section title)` . The name of the command simply changes to `subsection` or `subsubsection` as required. These commands give a label `<section-name>` and title `(section title)` to the section.

A special case of a page group is the main page group. The Doxygen command `\mainpage` within a comment block places the documentation within that block on the **Index** page of the generated documentation. You can refer to the main page using `\ref index` (if the treeview is disabled, otherwise you should use `\ref main`).

Formulas

Doxygen can include LaTeX formulas in the HTML and LaTeX output formats. Formulas can be included within the text of a single line, for example—$r = \sqrt{(x-x_0)^2 - (y-y_0)^2}$ or as a centered formula such as (from the Doxygen manual):

$$|I_2| = \left| \int_0^T \psi(t) \left\{ u(a,t) - \int_{\gamma(t)}^{a} \frac{d\theta}{k(\theta,t)} \int_a^{\theta} c(\xi) u_t(\xi,t)\, d\xi \right\} dt \right|$$

To include an inline formula the command `\f$` is used to begin and end the formula block, the above formula would appear in the documentation as `\f$ \sqrt{(x-x_0)^2-(y-y_0)^2} \f$`. A centered formula is enclosed by the Doxygen commands `\f[` to open the formula block and `\f]` to close it. Additional LaTeX equation environments, like `eqnarray` can be used with the commands `\f{eqnarray}` to open the formula block and `\f}` to close the block.

LaTeX Formulas

LaTeX formulas use a markup language with a rich set of tags which can be combined to typeset any formula. A brief sub-set of those commands is included here. Additional resources can be found on the web and in bookstores.

`x^{2}`	x^2; superscript	`x_{0}`	x_0; subscript
`\frac{x}{y}`	$\frac{x}{y}$	`\sqrt[n]{x}`	$\sqrt[n]{x}$
`\sum_{k=1}^n`	$\sum_{k=1}^{n}$	`\int_{a}^b dt`	$\int_a^b dt$
`\pi`	π; small Greek	`\Pi`	Π; capital Greek

Graphs

Doxygen can produce graphs, generated by the `dot` tool from Graphviz (http://www.graphviz.org/). Graph generation by Doxygen is normally performed automatically based on settings in the configuration file. These settings, with the exception of `DOT_PATH` are either YES or NO. These are *global* settings for the project. Graphs will only be generated if you have `dot` installed and `HAVE_DOT = YES`.

`HAVE_DOT`	signals that the `dot` tool is available.
`DOT_PATH`	path to`dot` tool, if not in `$PATH`.
`GRAPHICAL_HIERARCHY`	generate a graph of class heirarchy.
`CLASS_GRAPH`	generate inheritance graph for each documented class.
`INCLUDE_GRAPH`	generate dependency graph for each documented file.
`CALL_GRAPH`	generate call graph for each documented function or method.
`CALLER_GRAPH`	generate a graph indicating functions called by the documented function.
`COLLABORATION_GRAPH`	generate a graph showing inheritance and usage relationships between classes and structs.

You can also use the `dot` language syntax to generate a graph. Commands specific to `dot` are enclosed in the Doxygen command pair `\dot` and `\enddot`.

Special Commands

A listing Doxygen specific commands by category.

Text Formatting

`\a <word>`	*word* font face, equivalent to `\e`.
`\arg`	list item, equivalent to `\li`.
`\b <word>`	**word** font face.
`\c <word>`	`word` font face.
`\code`	starts a code block section.
`\endcode`	ends a `\code` section.
`\copydoc ref`	copies documenation from `ref`.
`\dot`	starts a dot graph block.
`\enddot`	ends a dot graph block.
`\e <word>`	*word* font face, equivalent to `\a`.
`\f$`	starts and ends an inline formula, i.e. $\sqrt{x}$.
`\f[`	starts a centered formula block.
`\f]`	ends a centered formula block.
`\image <format> <file> ``caption''`	places an image into the documentation with an optional caption
`\htmlonly`	starts a block for only HTML output.
`\endhtmlonly`	ends an HMTL only block.
`\n`	forces a new line.
`\p <word>`	`word` font face.
`\verbatim`	starts a block included as verbatim text.
`\endverbatim`	ends a verbatim text block.

Structural Indicators

`\addtogroup <l> (t)`	adds to a group, enclose in open-close pairs.
`\callgraph`	generates a call graph for function or method
`\callergraph`	generates a caller graph for function or method
`\category`	documentation block for a class category (Objective-C only).
`\class <c> [<f>] [<n>]`	documents the class `c`, header file `f` and header name `n` can be included.
`\def <name>`	documents the `name #define` macro
`\defgroup <l> ``<t>''`	defines a module group with a label `l` and title `t`.
`\enum <name>`	documents the `name` enumeration
`\example <file>`	a documentation block for an example contained in `file`.
`\file <name>`	a documentation block for file `name`
`\fn <sig>`	documents the function with signature `sig`
`\hideinitializer`	hides the default value of a `#define`
`\ingroup <l1> [<l2>]`	adds documentation to the `l1` group, multiple groups can be used.
`\interface <n>`	documentation block for interface `n`
`\internal`	all text following this is suppressed to the end of the comment block.
`\mainpage (t)`	the main or index page documentation block.
`\name (header)`	a member group documentation block.
`\namespace <n>`	a documenation block for namesapce `n`
`\nosubgrouping`	turns off sub-gropuing for a class.
`\overload (s)`	used to document an overloaded function with signature `s`
`\package <n>)`	a documentation block for package `n`
`\page <n> (t)`	indicates a page group documentation block with label `n` and title `t`
`\property (q)`	documentation for global or class property `q`
`\protocol (f)`	documentation block for a protocol (Objective-C only).
`\relates <n>`	used to relate non-member functions to classes
`\relatesalso <n>`	similar to `\relates`
`\showinitializer`	shows the default value of a `#define`
`\struct <n>`	documentation block for struct `n`
`\typedef (t)`	documentation block for typedef `t`
`\union <n>`	documentation block for union `n`
`\var (v)`	documentation block for variable `v`
`\weakgroup <n>`	equivalent to `\addtogroup` but has lower priority when resolving group conflicts.

Section Indicators

`\attention {...}`	starts an **Attention** paragraph.
`\author {loa}`	includes the list of authors `loa`.
`\brief {...}`	a brief description of the element.
`\bug {...}`	a bug description.
`\cond <sec>`	begins a conditional section.
`\date {...}`	
`\deprecated {...}`	starts a **Deprecated** paragraph.
`\else`	additional clause to a `\if` command
`\elseif <sec>`	additional clause to a `\cond` command.
`\endcond`	ends a conditional section.
`\endif`	ends an `\if` clause.
`\exception <e>`	starts an exception description for `e`
`\if <sec>`	starts a conditional section.
`\ifnot <sec>`	starts a conditional section.
`\invariant {...}`	starts a description of an invariant.
`\note {...}`	starts a **Note** paragraph.
`\par (t) {...}`	starts a paragraph with and optional title (t).
`\param <p> {...}`	starts a description of parameter `p`.
`\post {...}`	starts a post-condition description.
`\pre {...}`	starts a pre-condition description.
`\remarks {...}`	starts a **Remarks** paragraph.
`\return {...}`	starts a description of return values.
`\retval <f> {...}`	describes a return value for function `f`
`\sa {ref}`	starts a **See Also** paragraph.
`\see {ref}`	equivalent to `\sa`.
`\since {...}`	describes since when an entity was available.
`\test {...}`	starts a test case description paragraph.
`\throw <e> {...}`	equivalent to `exception`
`\todo {...}`	starts a **To Do** paragraph.
`\version {...}`	starts a paragraph where version strings can be entered.
`\warning {...}`	starts a paragraph for warning messages.
`\xrefitem <k> ``(h)'' {.}`	creats a cross-refernced paragraph.

Link Indicators

`\addindex (t)`	adds `t` to the LaTeX index.
`\anchor (w)`	places invisible anchor `w` in the document.
`\endlink`	ends a link started with `\link`
`\link <n> {...}`	creates link to `n` with user specified link text.
`\ref <n> ``(t)''`	creates a reference to `n` with text `t`.
`\subpage <n> ``(t)''`	creates a sub-page of name `n`.
`\section <l> (t)`	creates a section on a page with a label `l` and title `t`.
`\subsection <l> (t)`	creates a sub-section on a page with a label `l` and title `t`.
`\paragraph <n> (t)`	creates a paragraph on a page with a label `l` and title `t`.

2006 - Paul W. Joireman, joireman@fnal.gov
Based on the LaTeX 2_ε Cheat Sheet by Winston Chang
For more detailed information consult the Doxygen manual
http://www.stack.nl/ dimitri/doxygen/manual.html
`$Revision: 1.0 $, $Date: 2006/10/10 $.`