

Doxygen

zakaria.meliani

March 2019

Contents

1	Introduction	2
1.1	Comments in code source	2
1.2	Linux	2
1.3	mac	2
1.4	Windows	2
1.5	GIT repository	2
2	Generate documentation	3
2.1	C code	3
2.2	File description	3
2.3	Class description	5
2.4	Stucture	7
2.5	Add a code	7
2.6	Fortran code	7
2.7	Type description	8
3	Doxygen configuration file	9
3.1	Draw graphics	10
3.2	Use Markdown	10
4	Exercises	11
4.1	Fortran case	11
4.2	C case	14
5	Reference	14
A	Configuration file in Fortran case	15

Abstract

All files can be download from [courses and exercise](#)

1 Introduction

Doxygen is a documentation system for C, C++, Fortran, Python, and others languages. It allows to generate documentation for you developments:

- from comments inserted in the code source.
- from code structure it self.

The generated documentation can be used with browser (in HTML) and/or an off-line reference manual (in L^AT_EX), RTF, DOC files.

1.1 Comments in code source

There are two types of comments in code sources. The first kind are comments at the beginning of each file. It describes the file and lists the authors and known bugs. The second kind of comments are more local within the file that describe your procedures/function associated arguments and parameters and class/type/structure in use within the file.

1.2 Linux

```
sudo apt-get install Graphviz
sudo apt-get install doxygen doxygen-gui
```

1.3 mac

```
brew install Graphviz
brew install doxygen
```

1.4 Windows

from this link: <http://doxygen.nl/files/doxygen-1.8.15-setup.exe>

1.5 GIT repository

If you the following : g++, python, cmake are installed. You can install with git repository git clone <https://github.com/doxygen/doxygen.git>

```
cd doxygen
mkdir build
cd build
cmake -G "Unix Makefiles" ..
make
make install
```

2 Generate documentation

Doxygen documentation of any “object” (file, function, class, ...) consists of brief and detailed description using specific Doxygen comments which are different from others comments

2.1 C code

```
2 /**  
   * This is multilignes comments  
   * for C and C++ language  
4 */
```

Doxygen C code comment 1

Or

```
2 /*!  
   * You can use this forms  
   * for C and C++ language  
4 */
```

Doxygen C code comment 2

Or

```
2 /*!  
   You can use this forms  
   /*! for C and C++ language  
4 /*!
```

Doxygen C code comment 3

Or

```
2 ///  
   You can use this forms  
   /// for C and C++ language  
4 ///
```

Doxygen C code comment 4

Doxygen supports also in-line comments, both brief and detailed kind; these blocks can only be used to document members and parameters!

```
int i_param = 5; /** on the same line**/
```

Doxygen C code line comment

2.2 File description

Each file should be describe as follow

```

1000  /*
1002  FILE HEADER
1004  TITLE      : project name
1005  PROJECT    : sub-project name
1006  MODULE     : name of the module or program
1007  URL        : ...
1008  AFFILIATION : Observatoire
1009  DATE       : ...
1010  REVISION   : ... V0.8
1011  */
1012  /*!
1013   * @file [filename]
1014   *
1015   * @author [your name]
1016   * @date
1017   *
1018   * @brief [brief description]
1019   * @details [detailed description but not need]
1020   * @section
1021   * @f[
1022   *   \vec{v}\cdot\vec{\nabla}\vec{v}
1023   * @f]
1024   * @see [https://URL]
1025   * @bug no bug
1026   * @return [Hello word]
1027  */
1028  int main(int argc, char** argv)
1029  {
1030      ...
1031  }

```

codes/C_codes/Simple_files/header_file_C.c

1. First you should use @file to attribute name to file documentation file

```
@file [ file name ]
```

2. The authors names/institutions is set using

```

/**
 * @author [ authors name ]
 * ## Institution
 * Observatoire de Paris
 * ## adress
 * Meudon
 */

```

Here we use Markdown to institution and address in bold

3. At the main program file header it is possible to impose it as the main page by adding

```
!> @mainpage [ add title ]
```

2.3 Class description

It important when use a class to declare it with

1. **@class** [class name]
and
2. **@enum** to document an enumeration type.

```
1000 /**
1001  * @file      class.C
1002  * @author     MELIANI
1003  * @version    1.0
1004  * @date       01 Mars 2019
1005  * @brief      TP du cours Doxygen
1006  *
1007  * @details    Ce programme permet de mettre en pratique
1008  *             les notions vues pendant le cours
1009  *             Calcul de la documentation de class en C
1010  *
1011  * @todo
1012  * plus generale
1013  *
1014  * @mainpage   Cours Doxygen
1015  *
1016  * @section    intro Introduction
1017  *             This code it use to learn Doxygen with fortran
1018  * @section    optims Optimisations
1019  *             \subsection loops Les boucles
1020  *             blablabla
1021  *             \subsection cstes Les constantes
1022  *             blablabla
1023  */
1024
1025 class TheClass
1026 {
1027     public:
1028
1029         /**
1030          * An enum.
1031          * More detailed enum description.
1032          */
1033
1034         enum TEnum {
1035             TVal1, /**< enum value TVal1. */
1036             TVal2, /**< enum value TVal2. */
1037             TVal3  /**< enum value TVal3. */
1038         }
1039         *enumPtr, /**< enum pointer. Details. */
1040         enumVar;  /**< enum variable. Details. */
1041
1042         /**
1043          * A constructor.
1044          * A more elaborate description of the constructor.
1045          */
1046         TheClass();
```

```

1048     /**
1050      * A destructor.
1052      * A more elaborate description of the destructor.
1054      */
1055     ~TheClass();

1056     /**
1057      * a normal member taking two arguments and returning an
1058      integer value.
1059      * @param a an integer argument.
1060      * @param s a constant character pointer.
1061      * @see TheClass()
1062      * @see ~TheClass()
1063      * @see testMeToo()
1064      * @see publicVar()
1065      * @return The test results
1066      */
1067     int AMEMEMBER(int a, const char *s);

1068     /**
1069      * A pure virtual member.
1070      * @see testMe()
1071      * @param c1 the first argument.
1072      * @param c2 the second argument.
1073      */
1074     virtual void testMeToo(char c1, char c2) = 0;

1075     /**
1076      * a public variable.
1077      * Details.
1078      */
1079     int publicVar;

1080     /**
1081      * a function variable.
1082      * Details.
1083      */
1084     int (*handler)(int a, int b);
};

```

codes/C_codes/Simple_files/class.C

2.4 Structure

```
1000 /**
1001  * @struct Str_t
1002  * @brief Objet cha ne de caract res .
1003  *
1004  * Str_t est un petit objet de gestion de cha nes de caract res .
1005  * La cha ne se termine obligatoirement par un z ro de fin et l'
1006  * objet
1007  * connait la taille de cha ne contient !
1008  */
1009 typedef struct
1010 {
1011     char * sz; /*!< Cha ne avec caract re null de fin de cha ne
1012     . */
1013     size_t len; /*!< Taille de la cha ne sz sans compter le z ro
1014     de fin. */
1015 }
1016 Str_t;
```

codes/C_codes/Simple_files/structure.C

2.5 Add a code

It is possible to integrate fraction of an example of the code to

```
1000 /**
1001  * @code
1002  * for ( i = 0 ; i < 5 ; i++ ) { megaFunc(i) ; }
1003  * @endcode
1004  * /
```

codes/C_codes/Simple_files/code_endcode_use.C

2.6 Fortran code

- !> or !< to start a Doxygen comment
- !> or !! to continue a Doxygen comment

```
1000 !-----
1001 !          FILE HEADER
1002 !-----
1003 ! TITLE           : project name
1004 ! PROJECT         : sub-project name
1005 ! MODULE          : name of the module or program
1006 ! URL             : ...
1007 ! AFFILIATION     : Observatoire
1008 ! DATE            : ...
1009 ! REVISION        : ... V 0.8
1010 !> @file
```

```

1012 !> [filename]
!> @author [your name]
!> @date
1014 !> @brief [brief description]
!> @details [detailed description but not need]
1016 !> @section The source term in use
!> @f[
1018 !> \vec{v}\cdot\vec{\nabla}\vec{v}
!> @f]
1020 !> @return [Hello word]
!> @see [https://URL]
1022 !> @bug No known bugs.
!
1024 module mod_simulation
!working code :-)
1026 end module mod_simulation

```

codes/Fortran_codes/Simple_files/header_file.F.f90

At the main program file header it is possible to impose it as the main page by adding

```
!> @mainpage [ add title ]
```

2.7 Type description

To well describe a type, you should :

1. Indicate that is a type with

```
!> @typedef [ type name ]
```

2. To turn the type block into a header definition

```
!> @name [ type name ]
```

```

1000 !> @typedef The_constants
!! @name The_constants
1002 !! @code
!! type(The_constants) :: global_constant
1004 !! call global_constant%write
!!@endcode
1006 type The_constants
    private
1008     double precision :: Gravity !> Gravitational
    acceleration vector (default=(0 0 -9.81)) (m s-2)
    double precision :: light_speed = 2.99e10 !> Reference local
    sea level
1010     public
    double precision :: Mass
1012 contains
    procedure :: set_gravity !> procedure set_gravity
1014     procedure :: set_config !> procedure set_configuration
    procedure :: set_density !> procedure set_density
1016     procedure :: write => write_constants
end type

```


3 Doxygen configuration file

First generate a default **Doxygen** configuration file with command

doxygen -g

after edit the file Doxyfile.

1. set name to the project

```
PROJECT_NAME = "Observatoire de Paris - Doxygen"
```

2. set version number to the project

```
PROJECT_NUMBER = 1.0
```

3. add a logo if need

```
PROJECT_LOGO = "doc/webpage/Logos/PSL.pdf"
```

4. set output language

```
OUTPUT_LANGUAGE = French
```

5. indicate source files folder

```
INPUT = src
It possible many source folders
INPUT += src2
```

6. mkdir doc folder

- mkdir doc
- cd doc
- mkdir webpage

7. set the output folder

```
OUTPUT_DIRECTORY = doc/webpage
```

8. show the source files

```
SOURCE_BROWSER = YES
```

9. show comments inside files/ function/class/type

```
EXTRACT_ALL = YES
EXTRACT_STATIC = YES
EXTRACT_PRIVATE = YES
```

10. Generated HTML documentation will contain a main index with vertical navigation menus that are dynamically created via Javascript

```
HTML_DYNAMIC_MENUS = YES
```

3.1 Draw graphics

1. Use dot tool to draw figures such diagrams

```
HAVE_DOT = YES
```

2. The representation class (C,C++, Python), type (fortran)

```
HIDE_UNDOC_RELATIONS = NO
```

3. With UML-class diagram, set

```
ULM_LOOK = NO
```

4. Generate a call dependency graph for every global function or class method

```
CALL_GRAPH = YES
```

3.2 Use Markdown

1. To use markdow support, you should set

```
MARKDOWN_SUPPORT = YES
```

2. It is possible to specify a markdown page that is to be used as the main page

```
USE_MDFILE_AS_MAINPAGE = project_main.md
```

or using

```
INPUT += project_main.md
```

An example of project_main.md file,

```
1000 @mainpage          Cool Doxygen documentation Project
1002
1004 @author            Zakaria Meliani
1006 @brief             Pretty nice project.
1008
1009     1. Using Markdown
1010     2. Even equation
1011         - Works
1012         @f[
1013             |I_{-2}|=\left| \int_{-0}^T \psi(t)
1014                 \left\{
1015                     u(a,t)-
1016                     \int_{-\gamma(t)}^a
1017                     \frac{d\theta}{k(\theta,t)}
1018                     \int_a^\theta c(\xi)u_t(\xi,t)\,d\xi
1019                 \right\} dt
1020             \right|
1021         @f]
1022 @date:              Mars 12, 2019
```

```

1024 @version          0.9
1026 @section intro Introduction
1028     This code it use to learn Doxygen with fortran
1030 @details The files are in @ref https://www.overleaf.com/
        project/5c865e510cc99825a4010538
1032
1034 @todo compile it :- )
1036 @copyright https://github.com/example/ GNU General Public
        License 3
1038
@warning      Doxygen works well with C, C++, python

```

codes/Doxygen_config_files/project_main.md

4 Exercises

4.1 Fortran case

Add Doxygen comments to the following program

```

1000 ! Small test program
1001 ! base on "Modern Fortran in Practice" by Arjen Markus
1002 program test_objects
1004     use geometrical_objects
1006     implicit none
1008     type list_of_objects
1009         class(shape), pointer :: object
1010     end type
1011     type(list_of_objects), dimension(2) :: list
1012
1013     type(rectangle), target :: rect
1014     type(square), target :: sq
1016
1017     integer :: i
1018
1019     call rect%size( 1.0, 2.0 )
1020     call sq%size( 1.5 )
1022
1023     list(1)%object => rect
1024     list(2)%object => sq
1026
1027     do i = 1, size(list)
1028         write(*,*) 'Area: ', list(i)%object%get_area()
1029     enddo

```

```

1028 | end program test_objects

```

codes/Practical_works/Fortran/prog_geometry.f90

with associated module

```

1000 | ! classes_geom.f90 —
1001 | !   Classes of geometrical objects as an illustration
1002 | !   of object-oriented programming in Fortran
1003 | !
1004 | !   Example belonging to "Modern Fortran in Practice" by Arjen
1005 | !   Markus
1006 | !
1007 | !   This work is licensed under the Creative Commons Attribution
1008 | !   3.0 Unported License.
1009 | !   To view a copy of this license, visit http://creativecommons.org/licenses/by/3.0/
1010 | !   or send a letter to:
1011 | !   Creative Commons, 444 Castro Street, Suite 900, Mountain View
1012 | !   , California, 94041, USA.
1013 |
1014 | module geometrical_objects
1015 |
1016 |   implicit none
1017 |
1018 |   !
1019 |   ! General shape
1020 |   !
1021 |   type, abstract :: shape
1022 |     ! No data
1023 |     contains
1024 |       procedure(get_shape_area), deferred :: get_area
1025 |       !procedure :: size —
1026 |     does not work
1027 |   end type shape
1028 |
1029 |   abstract interface
1030 |     real function get_shape_area( this )
1031 |       import :: shape
1032 |       class(shape), intent(in) :: this
1033 |     end function get_shape_area
1034 |   end interface
1035 |
1036 |   !
1037 |   ! Rectangle
1038 |   !
1039 |   type, extends(shape) :: rectangle
1040 |     real :: width, height
1041 |     contains
1042 |       procedure :: get_area => get_rectangle_area
1043 |       procedure :: size => rectangle_size
1044 |   end type rectangle
1045 |
1046 |   !
1047 |   ! Square
1048 |   ! Note:

```

```

! square_size must have the same interface as its rectangle
parent!
!
1046 type, extends(rectangle) :: square
1048 contains
        procedure :: get_area => get_square_area
1050        procedure :: size    => square_size
        end type square
1052
contains
1054
!
1056 ! The various routines and functions we need
!
1058 real function get_rectangle_area( this )
        class(rectangle), intent(in) :: this
1060
        get_rectangle_area = this%width * this%height
1062
end function get_rectangle_area
1064
subroutine rectangle_size( this, width, height )
1066        class(rectangle), intent(inout) :: this
        real, intent(in)                :: width
1068        real, intent(in), optional      :: height

        this%width = width
        if ( present(height) ) then
1072            this%height = height
        else
1074            this%height = width
        endif
1076
end subroutine rectangle_size
1078
subroutine square_size( this, width, height )
1080        class(square), intent(inout) :: this
        real, intent(in)                :: width
1082        real, intent(in), optional      :: height ! Ignored

        this%width = width
        this%height = 0.0
1086
end subroutine square_size
1088
real function get_square_area( this )
1090        class(square), intent(in) :: this

        get_square_area = this%width ** 2
1092
end function get_square_area
1094
end module geometrical_objects
1096

```

codes/Practical_works/Fortran/mod_geometry.f90

4.2 C case

```
1000 // C# program to calculate
1001 // Volume and Surface area of cone
1002 using System;
1003
1004 class GFG
1005 {
1006     static float pi = 3.14159f;
1007
1008     // Function to calculate
1009     // Volume of cone
1010     public static float volume(float r,
1011                               float h)
1012     {
1013         return (float)1 / 3 * pi * h *
1014                r * r;
1015     }
1016
1017     // Function to calculate
1018     // Surface area of cone
1019     public static float surface-area(float r,
1020                                     float s)
1021     {
1022         return pi * r * s + pi * r * r;
1023     }
1024
1025     // Driver Code
1026     public static void Main()
1027     {
1028         float radius = 5;
1029         float slant_height = 13;
1030         float height = 12;
1031         //float vol, sur_area;
1032
1033         // Printing value of volume
1034         // and surface area
1035         Console.WriteLine("Volume Of Cone : ");
1036         Console.WriteLine(volume(radius,
1037                                 height));
1038
1039         Console.WriteLine("Surface Area Of Cone : ");
1040         Console.WriteLine(surface-area(radius,
1041                                     slant_height));
1042     }
1043 }
1044
1045 // This code is contributed by "vt_m"
```

codes/Practical-works/C/pro-volume.c

5 Reference

1. [Doxygen official website.](#)

2. Doxygen Documentation- Grenoble

A Configuration file in Fortran case

An example of edited Doxygen configuration file for source files within a folder **src**

```
1000 # Doxyfile 1.8.13
1002 # This file describes the settings to be used by the documentation
      system
      # doxygen (www.doxygen.org) for a project.
1004 #
      # All text after a double hash (##) is considered a comment and is
      placed in
1006 # front of the TAG it is preceding.
      #
1008 # All text after a single hash (#) is considered a comment and will
      be ignored.
      # The format is:
1010 # TAG = value [value, ...]
      # For lists, items can also be appended using:
1012 # TAG += value [value, ...]
      # Values that contain spaces should be placed between quotes (\
      \").
1014 #
      _____
1016 # Project related configuration options
      #
      _____
1018
      # This tag specifies the encoding used for all characters in the
      config file
1020 # that follow. The default is UTF-8 which is also the encoding used
      for all text
      # before the first occurrence of this tag. Doxygen uses libiconv (
      or the iconv
1022 # built into libc) for the transcoding. See http://www.gnu.org/
      software/libiconv
      # for the list of possible encodings.
1024 # The default value is: UTF-8.
1026 DOXYFILE.ENCODING      = UTF-8
1028 # The PROJECTNAME tag is a single word (or a sequence of words
      surrounded by
      # double-quotes, unless you are using Doxywizard) that should
      identify the
1030 # project for which the documentation is generated. This name is
      used in the
      # title of most generated pages and in a few other places.
1032 # The default value is: My Project.
```

```

1034 PROJECT_NAME           = "Observatoire de Paris – Code Fortran"
1036 # The PROJECT_NUMBER tag can be used to enter a project or revision
      # number. This
      # could be handy for archiving the generated documentation or if
      # some version
1038 # control system is used.
1040 PROJECT_NUMBER         = version 1.0
1042 # Using the PROJECT_BRIEF tag one can provide an optional one line
      # description
      # for a project that appears at the top of each page and should
      # give viewer a
1044 # quick idea about the purpose of the project. Keep the description
      # short.
1046 PROJECT_BRIEF          =
1048 # With the PROJECT_LOGO tag one can specify a logo or an icon that
      # is included
      # in the documentation. The maximum height of the logo should not
      # exceed 55
1050 # pixels and the maximum width should not exceed 200 pixels.
      # Doxygen will copy
      # the logo to the output directory.
1052 PROJECT_LOGO            = "doc/webpage/Logos/PSL.pdf"
1054 # The OUTPUT_DIRECTORY tag is used to specify the (relative or
      # absolute) path
1056 # into which the generated documentation will be written. If a
      # relative path is
      # entered, it will be relative to the location where doxygen was
      # started. If
1058 # left blank the current directory will be used.
1060 OUTPUT_DIRECTORY       = "doc/webpage"
1062 # If the CREATE_SUBDIRS tag is set to YES then doxygen will create
      # 4096 sub-
      # directories (in 2 levels) under the output directory of each
      # output format and
1064 # will distribute the generated files over these directories.
      # Enabling this
      # option can be useful when feeding doxygen a huge amount of source
      # files, where
1066 # putting all generated files in the same directory would otherwise
      # causes
      # performance problems for the file system.
1068 # The default value is: NO.
1070 CREATE_SUBDIRS          = NO
1072 # If the ALLOW_UNICODE_NAMES tag is set to YES, doxygen will allow
      # non-ASCII
      # characters to appear in the names of generated files. If set to

```



```

1074 NO, non-ASCII
# characters will be escaped, for example _xE3_x81_x84 will be used
# for Unicode
# U+3044.
1076 # The default value is: NO.

1078 ALLOW_UNICODE_NAMES      = NO

1080 # The OUTPUT_LANGUAGE tag is used to specify the language in which
# all
# documentation generated by doxygen is written. Doxygen will use
# this
1082 # information to generate all constant output in the proper
# language.
# Possible values are: Afrikaans, Arabic, Armenian, Brazilian,
# Catalan, Chinese,
1084 # Chinese-Traditional, Croatian, Czech, Danish, Dutch, English (
# United States),
# Esperanto, Farsi (Persian), Finnish, French, German, Greek,
# Hungarian,
1086 # Indonesian, Italian, Japanese, Japanese-en (Japanese with English
# messages),
# Korean, Korean-en (Korean with English messages), Latvian,
# Lithuanian,
1088 # Macedonian, Norwegian, Persian (Farsi), Polish, Portuguese,
# Romanian, Russian,
# Serbian, Serbian-Cyrillic, Slovak, Slovene, Spanish, Swedish,
# Turkish,
1090 # Ukrainian and Vietnamese.
# The default value is: English.

1092 OUTPUT_LANGUAGE           = English

1094 # If the BRIEF_MEMBER_DESC tag is set to YES, doxygen will include
# brief member
1096 # descriptions after the members that are listed in the file and
# class
# documentation (similar to Javadoc). Set to NO to disable this.
1098 # The default value is: YES.

1100 BRIEF_MEMBER_DESC         = YES

1102 # If the REPEAT_BRIEF tag is set to YES, doxygen will prepend the
# brief
# description of a member or function before the detailed
# description
1104 #
# Note: If both HIDE_UNDOC_MEMBERS and BRIEF_MEMBER_DESC are set to
# NO, the
1106 # brief descriptions will be completely suppressed.
# The default value is: YES.

1108 REPEAT_BRIEF               = YES

1110 # This tag implements a quasi-intelligent brief description
# abbreviator that is
1112 # used to form the text in various listings. Each string in this

```

```

    list, if found
    # as the leading text of the brief description, will be stripped
    # from the text
1114 # and the result, after processing the whole list, is used as the
    # annotated
    # text. Otherwise, the brief description is used as-is. If left
    # blank, the
1116 # following values are used ($name is automatically replaced with
    # the name of
    # the entity):The $name class, The $name widget, The $name file, is
    # , provides,
1118 # specifies, contains, represents, a, an and the.

1120 ABBREVIATE_BRIEF      = "The $name class" \
    "The $name widget" \
1122    "The $name file" \
    is \
1124    provides \
    specifies \
1126    contains \
    represents \
1128    a \
    an \
1130    the

1132 # If the ALWAYS_DETAILED_SEC and REPEAT_BRIEF tags are both set to
    # YES then
    # doxygen will generate a detailed section even if there is only a
    # brief
1134 # description.
    # The default value is: NO.

1136 ALWAYS_DETAILED_SEC   = YES

1138 # If the INLINE_INHERITED_MEMB tag is set to YES, doxygen will show
    # all
1140 # inherited members of a class in the documentation of that class
    # as if those
    # members were ordinary class members. Constructors, destructors
    # and assignment
1142 # operators of the base classes will not be shown.
    # The default value is: NO.

1144 INLINE_INHERITED_MEMB = NO

1146 # If the FULL_PATH_NAMES tag is set to YES, doxygen will prepend
    # the full path
1148 # before file name in the file list and in the header files. If
    # set to NO the
    # shortest path that makes the file name unique will be used
1150 # The default value is: YES.

1152 FULL_PATH_NAMES       = YES

1154 # The STRIP_FROM_PATH tag can be used to strip a user-defined part
    # of the path.
    # Stripping is only done if one of the specified strings matches

```

```

the left-hand
1156 # part of the path. The tag can be used to show relative paths in
the file list.
# If left blank the directory from which doxygen is run is used as
the path to
1158 # strip.
#
1160 # Note that you can specify absolute paths here, but also relative
paths, which
# will be relative from the directory where doxygen is started.
1162 # This tag requires that the tag FULLPATHNAMES is set to YES.

1164 STRIP_FROM_PATH          =

1166 # The STRIP_FROM_INC_PATH tag can be used to strip a user-defined
part of the
# path mentioned in the documentation of a class, which tells the
reader which
1168 # header file to include in order to use a class. If left blank
only the name of
# the header file containing the class definition is used.
Otherwise one should
1170 # specify the list of include paths that are normally passed to the
compiler
# using the -I flag.

1172 STRIP_FROM_INC_PATH      =

1174 # If the SHORTNAMES tag is set to YES, doxygen will generate much
shorter (but
1176 # less readable) file names. This can be useful if your file
systems doesn't
# support long names like on DOS, Mac, or CD-ROM.
1178 # The default value is: NO.

1180 SHORTNAMES              = NO

1182 # If the JAVADOC_AUTOBRIEF tag is set to YES then doxygen will
interpret the
# first line (until the first dot) of a Javadoc-style comment as
the brief
1184 # description. If set to NO, the Javadoc-style will behave just
like regular Qt-
# style comments (thus requiring an explicit @brief command for a
brief
1186 # description.)
# The default value is: NO.

1188 JAVADOC_AUTOBRIEF        = NO

1190 # If the QT_AUTOBRIEF tag is set to YES then doxygen will interpret
the first
1192 # line (until the first dot) of a Qt-style comment as the brief
description. If
# set to NO, the Qt-style will behave just like regular Qt-style
comments (thus
1194 # requiring an explicit \brief command for a brief description.)

```

```

1196 # The default value is: NO.
1198 QT_AUTOBRIEF                = NO
1198 # The MULTILINE_CPP_IS_BRIEF tag can be set to YES to make doxygen
1200 # multi-line C++ special comment block (i.e. a block of //! or ///
# comments) as
# a brief description. This used to be the default behavior. The
# new default is
1202 # to treat a multi-line C++ comment block as a detailed description
. Set this
# tag to YES if you prefer the old behavior instead.
1204 #
# Note that setting this tag to YES also means that rational rose
# comments are
1206 # not recognized any more.
# The default value is: NO.
1208 MULTILINE_CPP_IS_BRIEF = NO
1210 # If the INHERIT_DOCS tag is set to YES then an undocumented member
# inherits the
1212 # documentation from any documented member that it re-implements.
# The default value is: YES.
1214 INHERIT_DOCS                = YES
1216 # If the SEPARATE_MEMBER_PAGES tag is set to YES then doxygen will
# produce a new
1218 # page for each member. If set to NO, the documentation of a member
# will be part
# of the file/class/namespace that contains it.
1220 # The default value is: NO.
1222 SEPARATE_MEMBER_PAGES = NO
1224 # The TAB_SIZE tag can be used to set the number of spaces in a tab
. Doxygen
# uses this value to replace tabs by spaces in code fragments.
1226 # Minimum value: 1, maximum value: 16, default value: 4.
1228 TAB_SIZE                    = 8
1230 # This tag can be used to specify a number of aliases that act as
# commands in
# the documentation. An alias has the form:
1232 # name=value
# For example adding
1234 # "sideeffect=@par Side Effects:\n"
# will allow you to put the command \sideeffect (or @sideeffect) in
# the
1236 # documentation, which will result in a user-defined paragraph with
# heading
# "Side Effects:". You can put \n's in the value part of an alias
# to insert
1238 # newlines.

```

```

1240 ALIASES                                =
1242 # This tag can be used to specify a number of word-keyword mappings
      # (TCL only).
      # A mapping has the form "name=value". For example adding "class=
      itcl::class"
1244 # will allow you to use the command class in the itcl::class
      meaning.
1246 TCLSUBST                                =
1248 # Set the OPTIMIZE_OUTPUT_FOR_C tag to YES if your project consists
      of C sources
      # only. Doxygen will then generate output that is more tailored for
      C. For
1250 # instance, some of the names that are used will be different. The
      list of all
      # members will be omitted, etc.
1252 # The default value is: NO.
1254 OPTIMIZE_OUTPUT_FOR_C = NO
1256 # Set the OPTIMIZE_OUTPUT_JAVA tag to YES if your project consists
      of Java or
      # Python sources only. Doxygen will then generate output that is
      more tailored
1258 # for that language. For instance, namespaces will be presented as
      packages,
      # qualified scopes will look different, etc.
1260 # The default value is: NO.
1262 OPTIMIZE_OUTPUT_JAVA = NO
1264 # Set the OPTIMIZE_FOR_FORTRAN tag to YES if your project consists
      of Fortran
      # sources. Doxygen will then generate output that is tailored for
      Fortran.
1266 # The default value is: NO.
1268 OPTIMIZE_FOR_FORTRAN = NO
1270 # Set the OPTIMIZE_OUTPUT_VHDL tag to YES if your project consists
      of VHDL
      # sources. Doxygen will then generate output that is tailored for
      VHDL.
1272 # The default value is: NO.
1274 OPTIMIZE_OUTPUT_VHDL = NO
1276 # Doxygen selects the parser to use depending on the extension of
      the files it
      # parses. With this tag you can assign which parser to use for a
      given
1278 # extension. Doxygen has a built-in mapping, but you can override
      or extend it
      # using this tag. The format is ext=language, where ext is a file

```

```

extension, and
1280 # language is one of the parsers supported by doxygen: IDL, Java,
      Javascript,
      # C#, C, C++, D, PHP, Objective-C, Python, Fortran (fixed format
      Fortran:
1282 # FortranFixed, free formatted Fortran: FortranFree, unknown
      formatted Fortran:
      # Fortran. In the later case the parser tries to guess whether the
      code is fixed
1284 # or free formatted code, this is the default for Fortran type
      files), VHDL. For
      # instance to make doxygen treat .inc files as Fortran files (
      default is PHP),
1286 # and .f files as C (default is Fortran), use: inc=Fortran f=C.
      #
1288 # Note: For files without extension you can use no_extension as a
      placeholder.
      #
1290 # Note that for custom extensions you also need to set
      FILE_PATTERNS otherwise
      # the files are not read by doxygen.
1292
EXTENSION_MAPPING      =
1294
      # If the MARKDOWN_SUPPORT tag is enabled then doxygen pre-processes
      all comments
1296 # according to the Markdown format, which allows for more readable
      # documentation. See http://daringfireball.net/projects/markdown/
      for details.
1298 # The output of markdown processing is further processed by doxygen
      , so you can
      # mix doxygen, HTML, and XML commands with Markdown formatting.
      Disable only in
1300 # case of backward compatibilities issues.
      # The default value is: YES.
1302
MARKDOWN_SUPPORT      = YES
1304
      # When the TOC_INCLUDE_HEADINGS tag is set to a non-zero value, all
      headings up
1306 # to that level are automatically included in the table of contents
      , even if
      # they do not have an id attribute.
1308 # Note: This feature currently applies only to Markdown headings.
      # Minimum value: 0, maximum value: 99, default value: 0.
1310 # This tag requires that the tag MARKDOWN_SUPPORT is set to YES.
1312
TOC_INCLUDE_HEADINGS  = 0
1314
      # When enabled doxygen tries to link words that correspond to
      documented
      # classes, or namespaces to their corresponding documentation. Such
      a link can
1316 # be prevented in individual cases by putting a % sign in front of
      the word or
      # globally by setting AUTOLINK_SUPPORT to NO.
1318 # The default value is: YES.

```

```

1320 AUTOLINK_SUPPORT          = YES

1322 # If you use STL classes (i.e. std::string, std::vector, etc.) but
      # do not want
      # to include (a tag file for) the STL sources as input, then you
      # should set this
1324 # tag to YES in order to let doxygen match functions declarations
      # and
      # definitions whose arguments contain STL classes (e.g. func(std::
      # string);
1326 # versus func(std::string) {}). This also make the inheritance and
      # collaboration
      # diagrams that involve STL classes more complete and accurate.
1328 # The default value is: NO.

1330 BUILTIN_STL_SUPPORT       = NO

1332 # If you use Microsoft's C++/CLI language, you should set this
      # option to YES to
      # enable parsing support.
1334 # The default value is: NO.

1336 CPP_CLLSUPPORT             = NO

1338 # Set the SIP_SUPPORT tag to YES if your project consists of sip (
      # see:
      # http://www.riverbankcomputing.co.uk/software/sip/intro) sources
      # only. Doxygen
1340 # will parse them like normal C++ but will assume all classes use
      # public instead
      # of private inheritance when no explicit protection keyword is
      # present.
1342 # The default value is: NO.

1344 SIP_SUPPORT                = NO

1346 # For Microsoft's IDL there are propget and propput attributes to
      # indicate
      # getter and setter methods for a property. Setting this option to
      # YES will make
1348 # doxygen to replace the get and set methods by a property in the
      # documentation.
      # This will only work if the methods are indeed getting or setting
      # a simple
1350 # type. If this is not the case, or you want to show the methods
      # anyway, you
      # should set this option to NO.
1352 # The default value is: YES.

1354 IDLPROPERTY_SUPPORT        = YES

1356 # If member grouping is used in the documentation and the
      # DISTRIBUTE_GROUP_DOC
      # tag is set to YES then doxygen will reuse the documentation of
      # the first
1358 # member in the group (if any) for the other members of the group.

```

```

1360     By default
    # all members of a group must be documented explicitly.
    # The default value is: NO.

1362 DISTRIBUTE_GROUP_DOC    = NO

1364 # If one adds a struct or class to a group and this option is
    # enabled, then also
    # any nested class or struct is added to the same group. By default
    # this option
1366 # is disabled and one has to add nested compounds explicitly via \
    # ingroup.
    # The default value is: NO.

1368 GROUP_NESTED_COMPOUNDS = NO

1370 # Set the SUBGROUPING tag to YES to allow class member groups of
    # the same type
1372 # (for instance a group of public functions) to be put as a
    # subgroup of that
    # type (e.g. under the Public Functions section). Set it to NO to
    # prevent
1374 # subgrouping. Alternatively, this can be done per class using the
    # \nosubgrouping command.
1376 # The default value is: YES.

1378 SUBGROUPING              = YES

1380 # When the INLINE_GROUPED_CLASSES tag is set to YES, classes,
    # structs and unions
    # are shown inside the group in which they are included (e.g. using
    # \ingroup)
1382 # instead of on a separate page (for HTML and Man pages) or section
    # (for LaTeX
    # and RTF).
1384 #
    # Note that this feature does not work in combination with
1386 # SEPARATE_MEMBER_PAGES.
    # The default value is: NO.

1388 INLINE_GROUPED_CLASSES = NO

1390 # When the INLINE_SIMPLE_STRUCTS tag is set to YES, structs,
    # classes, and unions
1392 # with only public data fields or simple typedef fields will be
    # shown inline in
    # the documentation of the scope in which they are defined (i.e.
    # file,
1394 # namespace, or group documentation), provided this scope is
    # documented. If set
    # to NO, structs, classes, and unions are shown on a separate page
    # (for HTML and
1396 # Man pages) or section (for LaTeX and RTF).
    # The default value is: NO.

1398 INLINE_SIMPLE_STRUCTS    = NO

1400

```



```

# When TYPEDEF_HIDES_STRUCT tag is enabled, a typedef of a struct,
# union, or
1402 # enum is documented as struct, union, or enum with the name of the
# typedef. So
# typedef struct TypeS {} TypeT, will appear in the documentation
# as a struct
1404 # with name TypeT. When disabled the typedef will appear as a
# member of a file,
# namespace, or class. And the struct will be named TypeS. This can
# typically be
1406 # useful for C code in case the coding convention dictates that all
# compound
# types are typedef'ed and only the typedef is referenced, never
# the tag name.
1408 # The default value is: NO.

TYPEDEF_HIDES_STRUCT    = NO

1412 # The size of the symbol lookup cache can be set using
# LOOKUP_CACHE_SIZE. This
# cache is used to resolve symbols given their name and scope.
# Since this can be
1414 # an expensive process and often the same symbol appears multiple
# times in the
# code, doxygen keeps a cache of pre-resolved symbols. If the cache
# is too small
1416 # doxygen will become slower. If the cache is too large, memory is
# wasted. The
# cache size is given by this formula: 2^(16+LOOKUP_CACHE_SIZE).
# The valid range
1418 # is 0..9, the default is 0, corresponding to a cache size of
# 2^16=65536
# symbols. At the end of a run doxygen will report the cache usage
# and suggest
1420 # the optimal cache size from a speed point of view.
# Minimum value: 0, maximum value: 9, default value: 0.

1422 LOOKUP_CACHE_SIZE      = 0

1424 #
#
#
#
# Build related configuration options
#
#
#
#
# If the EXTRACT_ALL tag is set to YES, doxygen will assume all
# entities in
1430 # documentation are documented, even if no documentation was
# available. Private
# class members and static file members will be hidden unless the
1432 # EXTRACT_PRIVATE respectively EXTRACT_STATIC tags are set to YES.
# Note: This will also disable the warnings about undocumented
# members that are
1434 # normally produced when WARNINGS is set to YES.
# The default value is: NO.

```

```

1436 EXTRACT_ALL = YES
1438 # If the EXTRACT_PRIVATE tag is set to YES, all private members of
      a class will
1440 # be included in the documentation.
      # The default value is: NO.
1442 EXTRACT_PRIVATE = YES
1444 # If the EXTRACT_PACKAGE tag is set to YES, all members with
      package or internal
1446 # scope will be included in the documentation.
      # The default value is: NO.
1448 EXTRACT_STATIC = YES
1450 # If the EXTRACT_STATIC tag is set to YES, all static members of a
      file will be
1452 # included in the documentation.
      # The default value is: NO.
1454 EXTRACT_STATIC = NO
1456 # If the EXTRACT_LOCAL_CLASSES tag is set to YES, classes (and
      structs) defined
1458 # locally in source files will be included in the documentation. If
      set to NO,
      # only classes defined in header files are included. Does not have
      any effect
1460 # for Java sources.
      # The default value is: YES.
1462 EXTRACT_LOCAL_CLASSES = YES
1464 # This flag is only useful for Objective-C code. If set to YES,
      local methods,
1466 # which are defined in the implementation section but not in the
      interface are
      # included in the documentation. If set to NO, only methods in the
      interface are
1468 # included.
      # The default value is: NO.
1470 EXTRACT_LOCAL_METHODS = NO
1472 # If this flag is set to YES, the members of anonymous namespaces
      will be
1474 # extracted and appear in the documentation as a namespace called
      # 'anonymous_namespace{file}', where file will be replaced with the
      base name of
1476 # the file that contains the anonymous namespace. By default
      anonymous namespace
      # are hidden.
1478 # The default value is: NO.
1480 EXTRACT_ANON_NSPACES = NO

```

```

1482 # If the HIDEUNDOC.MEMBERS tag is set to YES, doxygen will hide
      all
      # undocumented members inside documented classes or files. If set
      to NO these
1484 # members will be included in the various overviews, but no
      documentation
      # section is generated. This option has no effect if EXTRACT_ALL is
      enabled.
1486 # The default value is: NO.

1488 HIDEUNDOC.MEMBERS      = NO

1490 # If the HIDEUNDOC.CLASSES tag is set to YES, doxygen will hide
      all
      # undocumented classes that are normally visible in the class
      hierarchy. If set
1492 # to NO, these classes will be included in the various overviews.
      This option
      # has no effect if EXTRACT_ALL is enabled.
1494 # The default value is: NO.

1496 HIDEUNDOC.CLASSES     = NO

1498 # If the HIDE_FRIEND_COMPOUNDS tag is set to YES, doxygen will hide
      all friend
      # (class|struct|union) declarations. If set to NO, these
      declarations will be
1500 # included in the documentation.
      # The default value is: NO.

1502 HIDE_FRIEND_COMPOUNDS  = NO

1504 # If the HIDE_IN_BODY_DOCS tag is set to YES, doxygen will hide any
1506 # documentation blocks found inside the body of a function. If set
      to NO, these
      # blocks will be appended to the function's detailed documentation
      block.
1508 # The default value is: NO.

1510 HIDE_IN_BODY_DOCS      = NO

1512 # The INTERNAL_DOCS tag determines if documentation that is typed
      after a
      # \internal command is included. If the tag is set to NO then the
      documentation
1514 # will be excluded. Set it to YES to include the internal
      documentation.
      # The default value is: NO.

1516 INTERNAL_DOCS          = NO

1518 # If the CASE_SENSE_NAMES tag is set to NO then doxygen will only
      generate file
1520 # names in lower-case letters. If set to YES, upper-case letters
      are also
      # allowed. This is useful if you have classes or files whose names

```

```

    only differ
1522 # in case and if your file system supports case sensitive file
    names. Windows
    # and Mac users are advised to set this option to NO.
1524 # The default value is: system dependent.

1526 CASE_SENSE_NAMES          = YES

1528 # If the HIDE_SCOPE_NAMES tag is set to NO then doxygen will show
    members with
    # their full class and namespace scopes in the documentation. If
    set to YES, the
1530 # scope will be hidden.
    # The default value is: NO.
1532 HIDE_SCOPE_NAMES          = NO

1534 # If the HIDE_COMPOUND_REFERENCE tag is set to NO (default) then
    doxygen will
1536 # append additional text to a page's title, such as Class Reference
    . If set to
    # YES the compound reference will be hidden.
1538 # The default value is: NO.

1540 HIDE_COMPOUND_REFERENCE= NO

1542 # If the SHOW_INCLUDE_FILES tag is set to YES then doxygen will put
    a list of
    # the files that are included by a file in the documentation of
    that file.
1544 # The default value is: YES.

1546 SHOW_INCLUDE_FILES        = YES

1548 # If the SHOW_GROUPED_MEMB_INC tag is set to YES then Doxygen will
    add for each
    # grouped member an include statement to the documentation, telling
    the reader
1550 # which file to include in order to use the member.
    # The default value is: NO.
1552 SHOW_GROUPED_MEMB_INC     = NO

1554 # If the FORCE_LOCAL_INCLUDES tag is set to YES then doxygen will
    list include
1556 # files with double quotes in the documentation rather than with
    sharp brackets.
    # The default value is: NO.
1558 FORCE_LOCAL_INCLUDES       = NO

1560 # If the INLINE_INFO tag is set to YES then a tag [inline] is
    inserted in the
1562 # documentation for inline members.
    # The default value is: YES.
1564 INLINE_INFO                = YES

```

```

1566 # If the SORT_MEMBER_DOCS tag is set to YES then doxygen will sort
      the
1568 # (detailed) documentation of file and class members alphabetically
      by member
      # name. If set to NO, the members will appear in declaration order.
1570 # The default value is: YES.

1572 SORT_MEMBER_DOCS          = YES

1574 # If the SORT_BRIEF_DOCS tag is set to YES then doxygen will sort
      the brief
      # descriptions of file , namespace and class members alphabetically
      by member
1576 # name. If set to NO, the members will appear in declaration order.
      Note that
      # this will also influence the order of the classes in the class
      list.
1578 # The default value is: NO.

1580 SORT_BRIEF_DOCS          = NO

1582 # If the SORT_MEMBERS_CTORS_1ST tag is set to YES then doxygen will
      sort the
      # (brief and detailed) documentation of class members so that
      constructors and
1584 # destructors are listed first. If set to NO the constructors will
      appear in the
      # respective orders defined by SORT_BRIEF_DOCS and SORT_MEMBER_DOCS
      .

1586 # Note: If SORT_BRIEF_DOCS is set to NO this option is ignored for
      sorting brief
      # member documentation.
1588 # Note: If SORT_MEMBER_DOCS is set to NO this option is ignored for
      sorting
      # detailed member documentation.
1590 # The default value is: NO.

1592 SORT_MEMBERS_CTORS_1ST = NO

1594 # If the SORT_GROUP_NAMES tag is set to YES then doxygen will sort
      the hierarchy
      # of group names into alphabetical order. If set to NO the group
      names will
1596 # appear in their defined order.
      # The default value is: NO.

1598 SORT_GROUP_NAMES          = NO

1600 # If the SORT_BY_SCOPE_NAME tag is set to YES, the class list will
      be sorted by
1602 # fully-qualified names, including namespaces. If set to NO, the
      class list will
      # be sorted only by class name, not including the namespace part.
1604 # Note: This option is not very useful if HIDE_SCOPE_NAMES is set
      to YES.
      # Note: This option applies only to the class list , not to the

```

```

    alphabetical
1606 # list.
    # The default value is: NO.
1608
    SORT.BY_SCOPE_NAME      = NO
1610
    # If the STRICT.PROTO.MATCHING option is enabled and doxygen fails
    # to do proper
1612 # type resolution of all parameters of a function it will reject a
    # match between
    # the prototype and the implementation of a member function even if
    # there is
1614 # only one candidate or it is obvious which candidate to choose by
    # doing a
    # simple string match. By disabling STRICT.PROTO.MATCHING doxygen
    # will still
1616 # accept a match between prototype and implementation in such cases
    #
    # The default value is: NO.
1618
    STRICT.PROTO.MATCHING   = NO
1620
    # The GENERATE.TODOLIST tag can be used to enable (YES) or disable
    # (NO) the todo
1622 # list. This list is created by putting \todo commands in the
    # documentation.
    # The default value is: YES.
1624
    GENERATE.TODOLIST       = YES
1626
    # The GENERATE.TESTLIST tag can be used to enable (YES) or disable
    # (NO) the test
1628 # list. This list is created by putting \test commands in the
    # documentation.
    # The default value is: YES.
1630
    GENERATE.TESTLIST       = YES
1632
    # The GENERATE.BUGLIST tag can be used to enable (YES) or disable (
    # NO) the bug
1634 # list. This list is created by putting \bug commands in the
    # documentation.
    # The default value is: YES.
1636
    GENERATE.BUGLIST        = YES
1638
    # The GENERATE.DEPRECATEDLIST tag can be used to enable (YES) or
    # disable (NO)
1640 # the deprecated list. This list is created by putting \deprecated
    # commands in
    # the documentation.
    # The default value is: YES.
1642
    GENERATE.DEPRECATEDLIST= YES
1644
    # The ENABLED.SECTIONS tag can be used to enable conditional
    # documentation

```

```

# sections, marked by \if <section_label> ... \endif and \cond <
# section_label>
1648 # ... \endcond blocks.

1650 ENABLED_SECTIONS =

1652 # The MAX_INITIALIZER_LINES tag determines the maximum number of
# lines that the
# initial value of a variable or macro / define can have for it to
# appear in the
1654 # documentation. If the initializer consists of more lines than
# specified here
# it will be hidden. Use a value of 0 to hide initializers
# completely. The
1656 # appearance of the value of individual variables and macros /
# defines can be
# controlled using \showinitializer or \hideinitializer command in
# the
1658 # documentation regardless of this setting.
# Minimum value: 0, maximum value: 10000, default value: 30.
1660
MAX_INITIALIZER_LINES = 30
1662
# Set the SHOW_USED_FILES tag to NO to disable the list of files
# generated at
1664 # the bottom of the documentation of classes and structs. If set to
# YES, the
# list will mention the files that were used to generate the
# documentation.
1666 # The default value is: YES.

1668 SHOW_USED_FILES = YES

1670 # Set the SHOW_FILES tag to NO to disable the generation of the
# Files page. This
# will remove the Files entry from the Quick Index and from the
# Folder Tree View
1672 # (if specified).
# The default value is: YES.
1674
SHOW_FILES = YES
1676
# Set the SHOW_NAMESPACES tag to NO to disable the generation of
# the Namespaces
1678 # page. This will remove the Namespaces entry from the Quick Index
# and from the
# Folder Tree View (if specified).
1680 # The default value is: YES.

1682 SHOW_NAMESPACES = YES

1684 # The FILE_VERSION_FILTER tag can be used to specify a program or
# script that
# doxygen should invoke to get the current version for each file (
# typically from
1686 # the version control system). Doxygen will invoke the program by
# executing (via

```

```

# popen()) the command command input-file, where command is the
# value of the
1688 # FILE.VERSION.FILTER tag, and input-file is the name of an input
# file provided
# by doxygen. Whatever the program writes to standard output is
# used as the file
1690 # version. For an example see the documentation.

1692 FILE.VERSION.FILTER      =

1694 # The LAYOUT_FILE tag can be used to specify a layout file which
# will be parsed
# by doxygen. The layout file controls the global structure of the
# generated
1696 # output files in an output format independent way. To create the
# layout file
# that represents doxygen's defaults, run doxygen with the -l
# option. You can
1698 # optionally specify a file name after the option, if omitted
# DoxygenLayout.xml
# will be used as the name of the layout file.

1700 #
# Note that if you run doxygen from a directory containing a file
# called
1702 # DoxygenLayout.xml, doxygen will parse it automatically even if
# the LAYOUT_FILE
# tag is left empty.

1704 LAYOUT_FILE              =

1706 # The CITE_BIB_FILES tag can be used to specify one or more bib
# files containing
1708 # the reference definitions. This must be a list of .bib files. The
# .bib
# extension is automatically appended if omitted. This requires the
# bibtex tool
1710 # to be installed. See also http://en.wikipedia.org/wiki/BibTeX for
# more info.
# For LaTeX the style of the bibliography can be controlled using
1712 # LATEX_BIB_STYLE. To use this feature you need bibtex and perl
# available in the
# search path. See also \cite for info how to create references.

1714 CITE_BIB_FILES          =

1716 #
# _____

1718 # Configuration options related to warning and progress messages
#
# _____

1720 #
# The QUIET tag can be used to turn on/off the messages that are
# generated to
1722 # standard output by doxygen. If QUIET is set to YES this implies
# that the

```



```

# messages are off.
1724 # The default value is: NO.

1726 QUIET = NO

1728 # The WARNINGS tag can be used to turn on/off the warning messages
# that are
# generated to standard error (stderr) by doxygen. If WARNINGS is
# set to YES
1730 # this implies that the warnings are on.
#
1732 # Tip: Turn warnings on while writing the documentation.
# The default value is: YES.
1734
1736 WARNINGS = YES

# If the WARN_IF_UNDOCUMENTED tag is set to YES then doxygen will
# generate
1738 # warnings for undocumented members. If EXTRACT_ALL is set to YES
# then this flag
# will automatically be disabled.
1740 # The default value is: YES.

1742 WARN_IF_UNDOCUMENTED = YES

1744 # If the WARN_IF_DOC_ERROR tag is set to YES, doxygen will generate
# warnings for
# potential errors in the documentation, such as not documenting
# some parameters
1746 # in a documented function, or documenting parameters that don't
# exist or using
# markup commands wrongly.
1748 # The default value is: YES.

1750 WARN_IF_DOC_ERROR = YES

1752 # This WARN_NO_PARAMDOC option can be enabled to get warnings for
# functions that
# are documented, but have no documentation for their parameters or
# return
1754 # value. If set to NO, doxygen will only warn about wrong or
# incomplete
# parameter documentation, but not about the absence of
# documentation.
1756 # The default value is: NO.

1758 WARN_NO_PARAMDOC = NO

1760 # If the WARN_AS_ERROR tag is set to YES then doxygen will
# immediately stop when
# a warning is encountered.
1762 # The default value is: NO.

1764 WARN_AS_ERROR = NO

1766 # The WARN_FORMAT tag determines the format of the warning messages
# that doxygen

```

```

# can produce. The string should contain the $file, $line, and
# $text tags, which
1768 # will be replaced by the file and line number from which the
# warning originated
# and the warning text. Optionally the format may contain $version,
# which will
1770 # be replaced by the version of the file (if it could be obtained
# via
# FILE_VERSION_FILTER)
1772 # The default value is: $file:$line: $text.

1774 WARNFORMAT                = "$file:$line: $text"

1776 # The WARN_LOGFILE tag can be used to specify a file to which
# warning and error
# messages should be written. If left blank the output is written
# to standard
1778 # error (stderr).

1780 WARN_LOGFILE              =

1782 #

# Configuration options related to the input files
1784 #

# The INPUT tag is used to specify the files and/or directories
# that contain
# documented source files. You may enter file names like myfile.cpp
# or
1788 # directories like /usr/src/myproject. Separate the files or
# directories with
# spaces. See also FILE_PATTERNS and EXTENSION_MAPPING
1790 # Note: If this tag is empty the current directory is searched.

1792 INPUT                      = "src_c"

1794 # This tag can be used to specify the character encoding of the
# source files
# that doxygen parses. Internally doxygen uses the UTF-8 encoding.
# Doxygen uses
1796 # libiconv (or the iconv built into libc) for the transcoding. See
# the libiconv
# documentation (see: http://www.gnu.org/software/libiconv) for the
# list of
1798 # possible encodings.
# The default value is: UTF-8.

1800 INPUT_ENCODING             = UTF-8

1802 # If the value of the INPUT tag contains directories, you can use
# the
1804 # FILE_PATTERNS tag to specify one or more wildcard patterns (like
# *.cpp and

```

```

# *.h) to filter out the source-files in the directories.
1806 #
# Note that for custom extensions or not directly supported
# extensions you also
1808 # need to set EXTENSION_MAPPING for the extension otherwise the
# files are not
# read by doxygen.
1810 #
# If left blank the following patterns are tested:*.c, *.cc, *.cxx,
# *.cpp,
1812 # *.c++, *.java, *.ii, *.ixx, *.ipp, *.i++, *.inl, *.idl, *.ddl, *.
# odl, *.h,
# *.hh, *.hxx, *.hpp, *.h++, *.cs, *.d, *.php, *.php4, *.php5, *.
# phtml, *.inc,
1814 # *.m, *.markdown, *.md, *.mm, *.dox, *.py, *.pyw, *.f90, *.f95, *.
# f03, *.f08,
# *.f, *.for, *.tcl, *.vhd, *.vhdl, *.ucf and *.qsf.
1816
FILE_PATTERNS = *.c \
1818 *.cc \
*.cxx \
1820 *.cpp \
*.c++ \
1822 *.java \
*.ii \
1824 *.ixx \
*.ipp \
1826 *.i++ \
*.inl \
1828 *.idl \
*.ddl \
1830 *.odl \
*.h \
1832 *.hh \
*.hxx \
1834 *.hpp \
*.h++ \
1836 *.cs \
*.d \
1838 *.php \
*.php4 \
1840 *.php5 \
*.phtml \
1842 *.inc \
*.m \
1844 *.markdown \
*.md \
1846 *.mm \
*.dox \
1848 *.py \
*.pyw \
1850 *.f90 \
*.f95 \
1852 *.f03 \
*.f08 \
1854 *.f \
*.for \

```

```

1856         *.tcl \
1857         *.vhd \
1858         *.vhdl \
1859         *.ucf \
1860         *.qsf

1862 # The RECURSIVE tag can be used to specify whether or not
1863     subdirectories should
1864 # be searched for input files as well.
1865 # The default value is: NO.

1866 RECURSIVE                = NO

1868 # The EXCLUDE tag can be used to specify files and/or directories
1869     that should be
1870 # excluded from the INPUT source files. This way you can easily
1871     exclude a
1872 # subdirectory from a directory tree whose root is specified with
1873     the INPUT tag.
1874 #
1875 # Note that relative paths are relative to the directory from which
1876     doxygen is
1877 # run.

1878 EXCLUDE                    =
1879 # The EXCLUDE_SYMLINKS tag can be used to select whether or not
1880     files or
1881 # directories that are symbolic links (a Unix file system feature)
1882     are excluded
1883 # from the input.
1884 # The default value is: NO.

1885 EXCLUDE_SYMLINKS          = NO

1886 # If the value of the INPUT tag contains directories, you can use
1887     the
1888 # EXCLUDE_PATTERNS tag to specify one or more wildcard patterns to
1889     exclude
1890 # certain files from those directories.
1891 #
1892 # Note that the wildcards are matched against the file with
1893     absolute path, so to
1894 # exclude all test directories for example use the pattern */test/*

1895 EXCLUDE_PATTERNS          =
1896 # The EXCLUDE_SYMBOLS tag can be used to specify one or more symbol
1897     names
1898 # (namespaces, classes, functions, etc.) that should be excluded
1899     from the
1900 # output. The symbol name can be a fully qualified name, a word, or
1901     if the
1902 # wildcard * is used, a substring. Examples: ANamespace, AClass,
1903     AClass::ANamespace, ANamespace::*Test
1904 #
1905 # Note that the wildcards are matched against the file with

```

```

1900         absolute path, so to
1900 # exclude all test directories use the pattern */test/*
1902 EXCLUDESYMBOLS          =
1904 # The EXAMPLEPATH tag can be used to specify one or more files or
        directories
        # that contain example code fragments that are included (see the \
        include
1906 # command).
1908 EXAMPLEPATH              =
1910 # If the value of the EXAMPLEPATH tag contains directories, you
        can use the
        # EXAMPLEPATTERNS tag to specify one or more wildcard pattern (
        like *.cpp and
1912 # *.h) to filter out the source-files in the directories. If left
        blank all
        # files are included.
1914 EXAMPLEPATTERNS          = *
1916 # If the EXAMPLERECURSIVE tag is set to YES then subdirectories
        will be
1918 # searched for input files to be used with the \include or \
        dontinclude commands
        # irrespective of the value of the RECURSIVE tag.
1920 # The default value is: NO.
1922 EXAMPLERECURSIVE         = NO
1924 # The IMAGEPATH tag can be used to specify one or more files or
        directories
        # that contain images that are to be included in the documentation
        (see the
1926 # \image command).
1928 IMAGEPATH                =
1930 # The INPUT.FILTER tag can be used to specify a program that
        doxygen should
        # invoke to filter for each input file. Doxygen will invoke the
        filter program
1932 # by executing (via popen()) the command:
        #
1934 # <filter> <input-file>
        #
1936 # where <filter> is the value of the INPUT.FILTER tag, and <input-
        file> is the
        # name of an input file. Doxygen will then use the output that the
        filter
1938 # program writes to standard output. If FILTER.PATTERNS is
        specified, this tag
        # will be ignored.
1940 #
        # Note that the filter must not add or remove lines; it is applied

```

```

    before the
1942 # code is scanned, but not when the output code is generated. If
    lines are added
    # or removed, the anchors will not be placed correctly.
1944 #
    # Note that for custom extensions or not directly supported
    extensions you also
1946 # need to set EXTENSION.MAPPING for the extension otherwise the
    files are not
    # properly processed by doxygen.
1948
INPUT_FILTER          =
1950
    # The FILTER_PATTERNS tag can be used to specify filters on a per
    file pattern
1952 # basis. Doxygen will compare the file name with each pattern and
    apply the
    # filter if there is a match. The filters are a list of the form:
    pattern=filter
1954 # (like *.cpp=my-cpp-filter). See INPUT_FILTER for further
    information on how
    # filters are used. If the FILTER_PATTERNS tag is empty or if none
    of the
1956 # patterns match the file name, INPUT_FILTER is applied.
    #
1958 # Note that for custom extensions or not directly supported
    extensions you also
    # need to set EXTENSION.MAPPING for the extension otherwise the
    files are not
1960 # properly processed by doxygen.

1962 FILTER_PATTERNS    =

1964 # If the FILTER_SOURCE_FILES tag is set to YES, the input filter (
    if set using
    # INPUT_FILTER) will also be used to filter the input files that
    are used for
1966 # producing the source files to browse (i.e. when SOURCE_BROWSER is
    set to YES).
    # The default value is: NO.
1968

FILTER_SOURCE_FILES   = NO
1970
    # The FILTER_SOURCE_PATTERNS tag can be used to specify source
    filters per file
1972 # pattern. A pattern will override the setting for FILTER_PATTERN (
    if any) and
    # it is also possible to disable source filtering for a specific
    pattern using
1974 # *.ext= (so without naming a filter).
    # This tag requires that the tag FILTER_SOURCE_FILES is set to YES.
1976

FILTER_SOURCE_PATTERNS =

1978
    # If the USE_MDFILE_AS_MAINPAGE tag refers to the name of a
    markdown file that
1980 # is part of the input, its contents will be placed on the main

```

```

    page
    # (index.html). This can be useful if you have a project on for
    # instance GitHub
1982 # and want to reuse the introduction page also for the doxygen
    # output.

1984 USE_MDFILE_AS_MAINPAGE =

1986 #
    _____

    # Configuration options related to source browsing
1988 #
    _____

1990 # If the SOURCE_BROWSER tag is set to YES then a list of source
    # files will be
    # generated. Documented entities will be cross-referenced with
    # these sources.
1992 #
    # Note: To get rid of all source code in the generated output, make
    # sure that
1994 # also VERBATIM_HEADERS is set to NO.
    # The default value is: NO.
1996
SOURCE_BROWSER                = NO
1998
    # Setting the INLINE_SOURCES tag to YES will include the body of
    # functions,
2000 # classes and enums directly into the documentation.
    # The default value is: NO.
2002
INLINE_SOURCES                = NO
2004
    # Setting the STRIP_CODE_COMMENTS tag to YES will instruct doxygen
    # to hide any
2006 # special comment blocks from generated source code fragments.
    # Normal C, C++ and
    # Fortran comments will always remain visible.
2008 # The default value is: YES.

2010 STRIP_CODE_COMMENTS        = YES

2012 # If the REFERENCED_BY_RELATION tag is set to YES then for each
    # documented
    # function all documented functions referencing it will be listed.
2014 # The default value is: NO.

2016 REFERENCED_BY_RELATION = NO

2018 # If the REFERENCES_RELATION tag is set to YES then for each
    # documented function
    # all documented entities called/used by that function will be
    # listed.
2020 # The default value is: NO.

```

```

2022 REFERENCES_RELATION      = NO

2024 # If the REFERENCES_LINK_SOURCE tag is set to YES and
      SOURCE_BROWSER tag is set
      # to YES then the hyperlinks from functions in REFERENCES_RELATION
      and
2026 # REFERENCED_BY_RELATION lists will link to the source code.
      Otherwise they will
      # link to the documentation.
2028 # The default value is: YES.

2030 REFERENCES_LINK_SOURCE = YES

2032 # If SOURCE_TOOLTIPS is enabled (the default) then hovering a
      hyperlink in the
      # source code will show a tooltip with additional information such
      as prototype,
2034 # brief description and links to the definition and documentation.
      Since this
      # will make the HTML file larger and loading of large files a bit
      slower, you
2036 # can opt to disable this feature.
      # The default value is: YES.
2038 # This tag requires that the tag SOURCE_BROWSER is set to YES.

2040 SOURCE_TOOLTIPS          = YES

2042 # If the USE_HTAGS tag is set to YES then the references to source
      code will
      # point to the HTML generated by the htags(1) tool instead of
      doxygen built-in
2044 # source browser. The htags tool is part of GNU's global source
      tagging system
      # (see http://www.gnu.org/software/global/global.html). You will
      need version
2046 # 4.8.6 or higher.
      #
2048 # To use it do the following:
      # - Install the latest version of global
2050 # - Enable SOURCE_BROWSER and USE_HTAGS in the config file
      # - Make sure the INPUT points to the root of the source tree
2052 # - Run doxygen as normal
      #
2054 # Doxygen will invoke htags (and that will in turn invoke gtags),
      so these
      # tools must be available from the command line (i.e. in the search
      path).
2056 #
      # The result: instead of the source browser generated by doxygen,
      the links to
2058 # source code will now point to the output of htags.
      # The default value is: NO.
2060 # This tag requires that the tag SOURCE_BROWSER is set to YES.

2062 USE_HTAGS                  = NO

2064 # If the VERBATIM_HEADERS tag is set the YES then doxygen will

```



```

    generate a
    # verbatim copy of the header file for each class for which an
    # include is
2066 # specified. Set to NO to disable this.
    # See also: Section \class.
2068 # The default value is: YES.

2070 VERBATIM_HEADERS          = YES

2072 # If the CLANG_ASSISTED_PARSING tag is set to YES then doxygen will
    # use the
    # clang parser (see: http://clang.llvm.org/) for more accurate
    # parsing at the
2074 # cost of reduced performance. This can be particularly helpful
    # with template
    # rich C++ code for which doxygen's built-in parser lacks the
    # necessary type
2076 # information.
    # Note: The availability of this option depends on whether or not
    # doxygen was
2078 # generated with the -Duse-libclang=ON option for CMake.
    # The default value is: NO.

2080 CLANG_ASSISTED_PARSING = NO

2082 # If clang assisted parsing is enabled you can provide the compiler
    # with command
2084 # line options that you would normally use when invoking the
    # compiler. Note that
    # the include paths will already be set by doxygen for the files
    # and directories
2086 # specified with INPUT and INCLUDE_PATH.
    # This tag requires that the tag CLANG_ASSISTED_PARSING is set to
    # YES.

2088 CLANG_OPTIONS              =
2090 #
    _____

2092 # Configuration options related to the alphabetical class index
    #
    _____

2094 # If the ALPHABETICAL_INDEX tag is set to YES, an alphabetical
    # index of all
2096 # compounds will be generated. Enable this if the project contains
    # a lot of
    # classes, structs, unions or interfaces.
2098 # The default value is: YES.

2100 ALPHABETICAL_INDEX        = YES

2102 # The COLS_IN_ALPHA_INDEX tag can be used to specify the number of
    # columns in
    # which the alphabetical index list will be split.

```

```

2104 # Minimum value: 1, maximum value: 20, default value: 5.
2105 # This tag requires that the tag ALPHABETICALINDEX is set to YES.
2106 COLS_IN_ALPHA_INDEX    = 5
2107
2108 # In case all classes in a project start with a common prefix, all
2109 # classes will
2110 # be put under the same header in the alphabetical index. The
2111 # IGNORE_PREFIX tag
2112 # can be used to specify a prefix (or a list of prefixes) that
2113 # should be ignored
2114 # while generating the index headers.
2115 # This tag requires that the tag ALPHABETICALINDEX is set to YES.
2116 IGNORE_PREFIX          =
2117 #
2118 # Configuration options related to the HTML output
2119 #
2120 # If the GENERATE_HTML tag is set to YES, doxygen will generate
2121 # HTML output
2122 # The default value is: YES.
2123 GENERATE_HTML           = YES
2124
2125 # The HTML_OUTPUT tag is used to specify where the HTML docs will
2126 # be put. If a
2127 # relative path is entered the value of OUTPUT_DIRECTORY will be
2128 # put in front of
2129 # it.
2130 # The default directory is: html.
2131 # This tag requires that the tag GENERATE_HTML is set to YES.
2132 HTML_OUTPUT             = html
2133
2134 # The HTML_FILE_EXTENSION tag can be used to specify the file
2135 # extension for each
2136 # generated HTML page (for example: .htm, .php, .asp).
2137 # The default value is: .html.
2138 # This tag requires that the tag GENERATE_HTML is set to YES.
2139 HTML_FILE_EXTENSION     = .html
2140
2141 # The HTML_HEADER tag can be used to specify a user-defined HTML
2142 # header file for
2143 # each generated HTML page. If the tag is left blank doxygen will
2144 # generate a
2145 # standard header.
2146 #
2147 # To get valid HTML the header file that includes any scripts and
2148 # style sheets
2149 # that doxygen needs, which is dependent on the configuration

```

```

options used (e.g.
# the setting GENERATE_TREEVIEW). It is highly recommended to start
  with a
2148 # default header using
# doxygen -w html new_header.html new_footer.html new_stylesheet.
  css
2150 # YourConfigFile
# and then modify the file new_header.html. See also section "
  Doxygen usage"
2152 # for information on how to generate the default header that
  doxygen normally
# uses.
2154 # Note: The header is subject to change so you typically have to
  regenerate the
# default header when upgrading to a newer version of doxygen. For
  a description
2156 # of the possible markers and block names see the documentation.
# This tag requires that the tag GENERATE_HTML is set to YES.
2158
HTMLHEADER          =
2160
# The HTMLFOOTER tag can be used to specify a user-defined HTML
  footer for each
2162 # generated HTML page. If the tag is left blank doxygen will
  generate a standard
# footer. See HTMLHEADER for more information on how to generate a
  default
2164 # footer and what special commands can be used inside the footer.
  See also
# section "Doxygen usage" for information on how to generate the
  default footer
2166 # that doxygen normally uses.
# This tag requires that the tag GENERATE_HTML is set to YES.
2168
HTMLFOOTER          =
2170
# The HTMLSTYLE SHEET tag can be used to specify a user-defined
  cascading style
2172 # sheet that is used by each HTML page. It can be used to fine-tune
  the look of
# the HTML output. If left blank doxygen will generate a default
  style sheet.
2174 # See also section "Doxygen usage" for information on how to
  generate the style
# sheet that doxygen normally uses.
2176 # Note: It is recommended to use HTML_EXTRA_STYLE SHEET instead of
  this tag, as
# it is more robust and this tag (HTML_STYLE SHEET) will in the
  future become
2178 # obsolete.
# This tag requires that the tag GENERATE_HTML is set to YES.
2180
HTMLSTYLE SHEET      =
2182
# The HTML_EXTRA_STYLE SHEET tag can be used to specify additional
  user-defined
2184 # cascading style sheets that are included after the standard style

```

```

    sheets
    # created by doxygen. Using this option one can overrule certain
    style aspects.
2186 # This is preferred over using HTMLSTYLESHEET since it does not
    replace the
    # standard style sheet and is therefore more robust against future
    updates.
2188 # Doxygen will copy the style sheet files to the output directory.
    # Note: The order of the extra style sheet files is of importance (
    e.g. the last
2190 # style sheet in the list overrules the setting of the previous
    ones in the
    # list). For an example see the documentation.
2192 # This tag requires that the tag GENERATEHTML is set to YES.

2194 HTML_EXTRA_STYLESHEET =

2196 # The HTML_EXTRA_FILES tag can be used to specify one or more extra
    images or
    # other source files which should be copied to the HTML output
    directory. Note
2198 # that these files will be copied to the base HTML output directory
    . Use the
    # $relpath^ marker in the HTMLHEADER and/or HTMLFOOTER files to
    load these
2200 # files. In the HTMLSTYLESHEET file, use the file name only. Also
    note that the
    # files will be copied as-is; there are no commands or markers
    available.
2202 # This tag requires that the tag GENERATEHTML is set to YES.

2204 HTML_EXTRA_FILES =

2206 # The HTML_COLORSTYLE_HUE tag controls the color of the HTML output
    . Doxygen
    # will adjust the colors in the style sheet and background images
    according to
2208 # this color. Hue is specified as an angle on a colorwheel, see
    # http://en.wikipedia.org/wiki/Hue for more information. For
    instance the value
2210 # 0 represents red, 60 is yellow, 120 is green, 180 is cyan, 240 is
    blue, 300
    # purple, and 360 is red again.
2212 # Minimum value: 0, maximum value: 359, default value: 220.
    # This tag requires that the tag GENERATEHTML is set to YES.

2214 HTML_COLORSTYLE_HUE = 220

2216 # The HTML_COLORSTYLE_SAT tag controls the purity (or saturation)
    of the colors
2218 # in the HTML output. For a value of 0 the output will use
    grayscales only. A
    # value of 255 will produce the most vivid colors.
2220 # Minimum value: 0, maximum value: 255, default value: 100.
    # This tag requires that the tag GENERATEHTML is set to YES.

2222 HTML_COLORSTYLE_SAT = 100

```

```

2224 # The HTMLCOLORSTYLEGAMMA tag controls the gamma correction
      # applied to the
2226 # luminance component of the colors in the HTML output. Values
      # below 100
      # gradually make the output lighter, whereas values above 100 make
      # the output
2228 # darker. The value divided by 100 is the actual gamma applied, so
      # 80 represents
      # a gamma of 0.8, The value 220 represents a gamma of 2.2, and 100
      # does not
2230 # change the gamma.
      # Minimum value: 40, maximum value: 240, default value: 80.
2232 # This tag requires that the tag GENERATEHTML is set to YES.

2234 HTMLCOLORSTYLEGAMMA = 80

2236 # If the HTMLTIMESTAMP tag is set to YES then the footer of each
      # generated HTML
      # page will contain the date and time when the page was generated.
      # Setting this
2238 # to YES can help to show when doxygen was last run and thus if the
      # documentation is up to date.
2240 # The default value is: NO.
      # This tag requires that the tag GENERATEHTML is set to YES.

2242 HTMLTIMESTAMP = NO

2244 # If the HTMLDYNAMICSECTIONS tag is set to YES then the generated
      # HTML
2246 # documentation will contain sections that can be hidden and shown
      # after the
      # page has loaded.
2248 # The default value is: NO.
      # This tag requires that the tag GENERATEHTML is set to YES.

2250 HTMLDYNAMICSECTIONS = NO

2252 # With HTMLINDEX_NUMENTRIES one can control the preferred number
      # of entries
2254 # shown in the various tree structured indices initially; the user
      # can expand
      # and collapse entries dynamically later on. Doxygen will expand
      # the tree to
2256 # such a level that at most the specified number of entries are
      # visible (unless
      # a fully collapsed tree already exceeds this amount). So setting
      # the number of
2258 # entries 1 will produce a full collapsed tree by default. 0 is a
      # special value
      # representing an infinite number of entries and will result in a
      # full expanded
2260 # tree by default.
      # Minimum value: 0, maximum value: 9999, default value: 100.
2262 # This tag requires that the tag GENERATEHTML is set to YES.

2264 HTMLINDEX_NUMENTRIES = 100

```

```

2266 # If the GENERATE_DOCSET tag is set to YES, additional index files
      # will be
      # generated that can be used as input for Apple's Xcode 3
      # integrated development
2268 # environment (see: http://developer.apple.com/tools/xcode/),
      # introduced with
      # OSX 10.5 (Leopard). To create a documentation set, doxygen will
      # generate a
2270 # Makefile in the HTML output directory. Running make will produce
      # the docset in
      # that directory and running make install will install the docset
      # in
2272 # ~/Library/Developer/Shared/Documentation/DocSets so that Xcode
      # will find it at
      # startup. See http://developer.apple.com/tools/
      # creatingdocsetswithdoxygen.html
2274 # for more information.
      # The default value is: NO.
2276 # This tag requires that the tag GENERATE_HTML is set to YES.

2278 GENERATE_DOCSET          = NO

2280 # This tag determines the name of the docset feed. A documentation
      # feed provides
      # an umbrella under which multiple documentation sets from a single
      # provider
2282 # (such as a company or product suite) can be grouped.
      # The default value is: Doxygen generated docs.
2284 # This tag requires that the tag GENERATE_DOCSET is set to YES.

2286 DOCSET_FEEDNAME          = "Doxygen generated docs"

2288 # This tag specifies a string that should uniquely identify the
      # documentation
      # set bundle. This should be a reverse domain-name style string, e.
      # g.
2290 # com.mycompany.MyDocSet. Doxygen will append .docset to the name.
      # The default value is: org.doxygen.Project.
2292 # This tag requires that the tag GENERATE_DOCSET is set to YES.

2294 DOCSET_BUNDLE_ID         = org.doxygen.Project

2296 # The DOCSET_PUBLISHER_ID tag specifies a string that should
      # uniquely identify
      # the documentation publisher. This should be a reverse domain-name
      # style
2298 # string, e.g. com.mycompany.MyDocSet.documentation.
      # The default value is: org.doxygen.Publisher.
2300 # This tag requires that the tag GENERATE_DOCSET is set to YES.

2302 DOCSET_PUBLISHER_ID      = org.doxygen.Publisher

2304 # The DOCSET_PUBLISHER_NAME tag identifies the documentation
      # publisher.
      # The default value is: Publisher.
2306 # This tag requires that the tag GENERATE_DOCSET is set to YES.

```

```

2308 DOCSET.PUBLISHER_NAME = Publisher

2310 # If the GENERATE_HTMLHELP tag is set to YES then doxygen generates
      three
      # additional HTML index files: index.hpp, index.hhc, and index.hhk.
      The
2312 # index.hpp is a project file that can be read by Microsoft's HTML
      Help Workshop
      # (see: http://www.microsoft.com/en-us/download/details.aspx?id=21138) on
2314 # Windows.
      #
2316 # The HTML Help Workshop contains a compiler that can convert all
      HTML output
      # generated by doxygen into a single compiled HTML file (.chm).
      Compiled HTML
2318 # files are now used as the Windows 98 help format, and will
      replace the old
      # Windows help format (.hlp) on all Windows platforms in the future
      . Compressed
2320 # HTML files also contain an index, a table of contents, and you
      can search for
      # words in the documentation. The HTML workshop also contains a
      viewer for
2322 # compressed HTML files.
      # The default value is: NO.
2324 # This tag requires that the tag GENERATE_HTML is set to YES.

2326 GENERATE_HTMLHELP = NO

2328 # The CHM_FILE tag can be used to specify the file name of the
      resulting .chm
      # file. You can add a path in front of the file if the result
      should not be
2330 # written to the html output directory.
      # This tag requires that the tag GENERATE_HTMLHELP is set to YES.
2332
      CHM_FILE =

2334
      # The HHC_LOCATION tag can be used to specify the location (
      absolute path
2336 # including file name) of the HTML help compiler (hhc.exe). If non-
      empty,
      # doxygen will try to run the HTML help compiler on the generated
      index.hpp.
2338 # The file has to be specified with full path.
      # This tag requires that the tag GENERATE_HTMLHELP is set to YES.
2340
      HHC_LOCATION =

2342
      # The GENERATE_CHI flag controls if a separate .chi index file is
      generated
2344 # (YES) or that it should be included in the master .chm file (NO).
      # The default value is: NO.
2346 # This tag requires that the tag GENERATE_HTMLHELP is set to YES.

```

```

2348 GENERATE_CHI                = NO
2350 # The CHM_INDEX_ENCODING is used to encode HtmlHelp index (hhk),
      content (hhc)
      # and project file content.
2352 # This tag requires that the tag GENERATE_HTMLHELP is set to YES.
2354 CHM_INDEX_ENCODING          =
2356 # The BINARY_TOC flag controls whether a binary table of contents
      is generated
      # (YES) or a normal table of contents (NO) in the .chm file.
      Furthermore it
2358 # enables the Previous and Next buttons.
      # The default value is: NO.
2360 # This tag requires that the tag GENERATE_HTMLHELP is set to YES.
2362 BINARY_TOC                   = NO
2364 # The TOC_EXPAND flag can be set to YES to add extra items for
      group members to
      # the table of contents of the HTML help documentation and to the
      tree view.
2366 # The default value is: NO.
      # This tag requires that the tag GENERATE_HTMLHELP is set to YES.
2368 TOC_EXPAND                    = NO
2370 # If the GENERATE_QHP tag is set to YES and both QHP_NAMESPACE and
2372 # QHP_VIRTUAL_FOLDER are set, an additional index file will be
      generated that
      # can be used as input for Qt's qhelpgenerator to generate a Qt
      Compressed Help
2374 # (.qch) of the generated HTML documentation.
      # The default value is: NO.
2376 # This tag requires that the tag GENERATE_HTML is set to YES.
2378 GENERATE_QHP                  = NO
2380 # If the QHG_LOCATION tag is specified, the QCH_FILE tag can be
      used to specify
      # the file name of the resulting .qch file. The path specified is
      relative to
2382 # the HTML output folder.
      # This tag requires that the tag GENERATE_QHP is set to YES.
2384 QCH_FILE                      =
2386 # The QHP_NAMESPACE tag specifies the namespace to use when
      generating Qt Help
2388 # Project output. For more information please see Qt Help Project /
      Namespace
      # (see: http://qt-project.org/doc/qt-4.8/qthelpproject.html#
      namespace).
2390 # The default value is: org.doxygen.Project.
      # This tag requires that the tag GENERATE_QHP is set to YES.
2392

```



```

2394 QHP_NAMESPACE                = org.doxygen.Project
2396 # The QHP_VIRTUAL_FOLDER tag specifies the namespace to use when
      # generating Qt
2396 # Help Project output. For more information please see Qt Help
      # Project / Virtual
      # Folders (see: http://qt-project.org/doc/qt-4.8/qthelpproject.html
2398 # #virtual-
      # folders).
      # The default value is: doc.
2400 # This tag requires that the tag GENERATE_QHP is set to YES.

2402 QHP_VIRTUAL_FOLDER          = doc

2404 # If the QHP_CUST_FILTER_NAME tag is set, it specifies the name of
      # a custom
      # filter to add. For more information please see Qt Help Project /
      # Custom
2406 # Filters (see: http://qt-project.org/doc/qt-4.8/qthelpproject.html
      # #custom-
      # filters).
2408 # This tag requires that the tag GENERATE_QHP is set to YES.

2410 QHP_CUST_FILTER_NAME        =

2412 # The QHP_CUST_FILTER_ATTRS tag specifies the list of the
      # attributes of the
      # custom filter to add. For more information please see Qt Help
      # Project / Custom
2414 # Filters (see: http://qt-project.org/doc/qt-4.8/qthelpproject.html
      # #custom-
      # filters).
2416 # This tag requires that the tag GENERATE_QHP is set to YES.

2418 QHP_CUST_FILTER_ATTRS      =

2420 # The QHP_SECT_FILTER_ATTRS tag specifies the list of the
      # attributes this
      # project's filter section matches. Qt Help Project / Filter
      # Attributes (see:
2422 # http://qt-project.org/doc/qt-4.8/qthelpproject.html#filter-attributes).
      # This tag requires that the tag GENERATE_QHP is set to YES.

2424 QHP_SECT_FILTER_ATTRS      =

2426 # The QHG_LOCATION tag can be used to specify the location of Qt's
2428 # qhelpgenerator. If non-empty doxygen will try to run
      # qhelpgenerator on the
      # generated .qhp file.
2430 # This tag requires that the tag GENERATE_QHP is set to YES.

2432 QHG_LOCATION                =

2434 # If the GENERATE_ECLIPSEHELP tag is set to YES, additional index
      # files will be
      # generated, together with the HTML files, they form an Eclipse

```

```

    help plugin. To
2436 # install this plugin and make it available under the help contents
    menu in
    # Eclipse, the contents of the directory containing the HTML and
    XML files needs
2438 # to be copied into the plugins directory of eclipse. The name of
    the directory
    # within the plugins directory should be the same as the
    ECLIPSE_DOC_ID value.
2440 # After copying Eclipse needs to be restarted before the help
    appears.
    # The default value is: NO.
2442 # This tag requires that the tag GENERATE_HTML is set to YES.

2444 GENERATE_ECLIPSE_HELP    = NO

2446 # A unique identifier for the Eclipse help plugin. When installing
    the plugin
    # the directory name containing the HTML and XML files should also
    have this
2448 # name. Each documentation set should have its own identifier.
    # The default value is: org.doxygen.Project.
2450 # This tag requires that the tag GENERATE_ECLIPSE_HELP is set to YES
    .

2452 ECLIPSE_DOC_ID          = org.doxygen.Project

2454 # If you want full control over the layout of the generated HTML
    pages it might
    # be necessary to disable the index and replace it with your own.
    The
2456 # DISABLE_INDEX tag can be used to turn on/off the condensed index
    (tabs) at top
    # of each HTML page. A value of NO enables the index and the value
    YES disables
2458 # it. Since the tabs in the index contain the same information as
    the navigation
    # tree, you can set this option to YES if you also set
    GENERATE_TREEVIEW to YES.
2460 # The default value is: NO.
    # This tag requires that the tag GENERATE_HTML is set to YES.

2462 DISABLE_INDEX            = NO

2464 # The GENERATE_TREEVIEW tag is used to specify whether a tree-like
    index
2466 # structure should be generated to display hierarchical information
    . If the tag
    # value is set to YES, a side panel will be generated containing a
    tree-like
2468 # index structure (just like the one that is generated for HTML
    Help). For this
    # to work a browser that supports JavaScript, DHTML, CSS and frames
    is required
2470 # (i.e. any modern browser). Windows users are probably better off
    using the
    # HTML help feature. Via custom style sheets (see

```

```

HTML_EXTRA_STYLESHEET) one can
2472 # further fine-tune the look of the index. As an example, the
      default style
      # sheet generated by doxygen has an example that shows how to put
      an image at
2474 # the root of the tree instead of the PROJECTNAME. Since the tree
      basically has
      # the same information as the tab index, you could consider setting
2476 # DISABLE_INDEX to YES when enabling this option.
      # The default value is: NO.
2478 # This tag requires that the tag GENERATE_HTML is set to YES.

2480 GENERATE_TREEVIEW          = NO

2482 # The ENUM_VALUES_PER_LINE tag can be used to set the number of
      enum values that
      # doxygen will group on one line in the generated HTML
      documentation.

2484 #
      # Note that a value of 0 will completely suppress the enum values
      from appearing
2486 # in the overview section.
      # Minimum value: 0, maximum value: 20, default value: 4.
2488 # This tag requires that the tag GENERATE_HTML is set to YES.

2490 ENUM_VALUES_PER_LINE      = 4

2492 # If the treeview is enabled (see GENERATE_TREEVIEW) then this tag
      can be used
      # to set the initial width (in pixels) of the frame in which the
      tree is shown.
2494 # Minimum value: 0, maximum value: 1500, default value: 250.
      # This tag requires that the tag GENERATE_HTML is set to YES.

2496 TREEVIEW_WIDTH             = 250

2498 # If the EXT_LINKS_IN_WINDOW option is set to YES, doxygen will
      open links to
2500 # external symbols imported via tag files in a separate window.
      # The default value is: NO.
2502 # This tag requires that the tag GENERATE_HTML is set to YES.

2504 EXT_LINKS_IN_WINDOW        = NO

2506 # Use this tag to change the font size of LaTeX formulas included
      as images in
      # the HTML documentation. When you change the font size after a
      successful
2508 # doxygen run you need to manually remove any form_*.png images
      from the HTML
      # output directory to force them to be regenerated.
2510 # Minimum value: 8, maximum value: 50, default value: 10.
      # This tag requires that the tag GENERATE_HTML is set to YES.

2512 FORMULA_FONT_SIZE          = 10

2514 # Use the FORMULA_TRANSPARENT tag to determine whether or not the

```

```

images
2516 # generated for formulas are transparent PNGs. Transparent PNGs are
      # not
      # supported properly for IE 6.0, but are supported on all modern
      # browsers.
2518 #
      # Note that when changing this option you need to delete any form-
      # *.png files in
2520 # the HTML output directory before the changes have effect.
      # The default value is: YES.
2522 # This tag requires that the tag GENERATEHTML is set to YES.

2524 FORMULA_TRANSPARENT      = YES

2526 # Enable the USEMATHJAX option to render LaTeX formulas using
      # MathJax (see
      # http://www.mathjax.org) which uses client side Javascript for the
      # rendering
2528 # instead of using pre-rendered bitmaps. Use this if you do not
      # have LaTeX
      # installed or if you want to formulas look prettier in the HTML
      # output. When
2530 # enabled you may also need to install MathJax separately and
      # configure the path
      # to it using the MATHJAX_RELPATH option.
2532 # The default value is: NO.
      # This tag requires that the tag GENERATEHTML is set to YES.

2534 USEMATHJAX                = NO

2536 # When MathJax is enabled you can set the default output format to
      # be used for
2538 # the MathJax output. See the MathJax site (see:
      # http://docs.mathjax.org/en/latest/output.html) for more details.
2540 # Possible values are: HTML-CSS (which is slower, but has the best
      # compatibility), NativeMML (i.e. MathML) and SVG.
2542 # The default value is: HTML-CSS.
      # This tag requires that the tag USEMATHJAX is set to YES.

2544 MATHJAX_FORMAT            = HTML-CSS

2546 # When MathJax is enabled you need to specify the location relative
      # to the HTML
2548 # output directory using the MATHJAX_RELPATH option. The
      # destination directory
      # should contain the MathJax.js script. For instance, if the
      # mathjax directory
2550 # is located at the same level as the HTML output directory, then
      # MATHJAX_RELPATH should be ../mathjax. The default value points to
      # the MathJax
2552 # Content Delivery Network so you can quickly see the result
      # without installing
      # MathJax. However, it is strongly recommended to install a local
      # copy of
2554 # MathJax from http://www.mathjax.org before deployment.
      # The default value is: http://cdn.mathjax.org/mathjax/latest.
2556 # This tag requires that the tag USEMATHJAX is set to YES.

```

```

2558 MATHJAX.RELPATH          = http://cdn.mathjax.org/mathjax/latest
2560 # The MATHJAX.EXTENSIONS tag can be used to specify one or more
      # MathJax
      # extension names that should be enabled during MathJax rendering.
      # For example
2562 # MATHJAX.EXTENSIONS = TeX/AMSmath TeX/AMSsymbols
      # This tag requires that the tag USE_MATHJAX is set to YES.
2564 MATHJAX.EXTENSIONS      =
2566 # The MATHJAX.CODEFILE tag can be used to specify a file with
      # javascript pieces
2568 # of code that will be used on startup of the MathJax code. See the
      # MathJax site
      # (see: http://docs.mathjax.org/en/latest/output.html) for more
      # details. For an
2570 # example see the documentation.
      # This tag requires that the tag USE_MATHJAX is set to YES.
2572 MATHJAX.CODEFILE        =
2574 # When the SEARCHENGINE tag is enabled doxygen will generate a
      # search box for
2576 # the HTML output. The underlying search engine uses javascript and
      # DHHTML and
      # should work on any modern browser. Note that when using HTML help
2578 # (GENERATEHTMLHELP), Qt help (GENERATEQHP), or docsets (
      # GENERATEDOCSET)
      # there is already a search function so this one should typically
      # be disabled.
2580 # For large projects the javascript based search engine can be slow
      # , then
      # enabling SERVER_BASED_SEARCH may provide a better solution. It is
      # possible to
2582 # search using the keyboard; to jump to the search box use <access
      # key> + S
      # (what the <access key> is depends on the OS and browser, but it
      # is typically
2584 # <CTRL>, <ALT>/<option>, or both). Inside the search box use the <
      # cursor down
      # key> to jump into the search results window, the results can be
      # navigated
2586 # using the <cursor keys>. Press <Enter> to select an item or <
      # escape> to cancel
      # the search. The filter options can be selected when the cursor is
      # inside the
2588 # search box by pressing <Shift>+<cursor down>. Also here use the <
      # cursor keys>
      # to select a filter and <Enter> or <escape> to activate or cancel
      # the filter
2590 # option.
      # The default value is: YES.
2592 # This tag requires that the tag GENERATEHTML is set to YES.
2594 SEARCHENGINE            = YES

```

```

2596 # When the SERVER_BASED_SEARCH tag is enabled the search engine
      # will be
      # implemented using a web server instead of a web client using
      # Javascript. There
2598 # are two flavors of web server based searching depending on the
      # EXTERNAL_SEARCH
      # setting. When disabled, doxygen will generate a PHP script for
      # searching and
2600 # an index file used by the script. When EXTERNAL_SEARCH is enabled
      # the indexing
      # and searching needs to be provided by external tools. See the
      # section
2602 # "External Indexing and Searching" for details.
      # The default value is: NO.
2604 # This tag requires that the tag SEARCHENGINE is set to YES.

2606 SERVER_BASED_SEARCH      = NO

2608 # When EXTERNAL_SEARCH tag is enabled doxygen will no longer
      # generate the PHP
      # script for searching. Instead the search results are written to
      # an XML file
2610 # which needs to be processed by an external indexer. Doxygen will
      # invoke an
      # external search engine pointed to by the SEARCHENGINE_URL option
      # to obtain the
2612 # search results.
      #
2614 # Doxygen ships with an example indexer (doxyindexer) and search
      # engine
      # (doxysearch.cgi) which are based on the open source search engine
      # library
2616 # Xapian (see: http://xapian.org/).
      #
2618 # See the section "External Indexing and Searching" for details.
      # The default value is: NO.
2620 # This tag requires that the tag SEARCHENGINE is set to YES.

2622 EXTERNAL_SEARCH          = NO

2624 # The SEARCHENGINE_URL should point to a search engine hosted by a
      # web server
      # which will return the search results when EXTERNAL_SEARCH is
      # enabled.
2626 #
      # Doxygen ships with an example indexer (doxyindexer) and search
      # engine
2628 # (doxysearch.cgi) which are based on the open source search engine
      # library
      # Xapian (see: http://xapian.org/). See the section "External
      # Indexing and
2630 # Searching" for details.
      # This tag requires that the tag SEARCHENGINE is set to YES.

2632 SEARCHENGINE_URL         =
2634

```

```

# When SERVER_BASED_SEARCH and EXTERNALSEARCH are both enabled the
#   unindexed
2636 # search data is written to a file for indexing by an external tool
#   . With the
# SEARCHDATA_FILE tag the name of this file can be specified.
2638 # The default file is: searchdata.xml.
# This tag requires that the tag SEARCHENGINE is set to YES.
2640 SEARCHDATA_FILE          = searchdata.xml
2642
# When SERVER_BASED_SEARCH and EXTERNALSEARCH are both enabled the
2644 # EXTERNAL_SEARCH_ID tag can be used as an identifier for the
#   project. This is
#   useful in combination with EXTRA_SEARCH_MAPPINGS to search
#   through multiple
2646 # projects and redirect the results back to the right project.
# This tag requires that the tag SEARCHENGINE is set to YES.
2648 EXTERNAL_SEARCH_ID      =
2650
# The EXTRA_SEARCH_MAPPINGS tag can be used to enable searching
#   through doxygen
2652 # projects other than the one defined by this configuration file ,
#   but that are
#   all added to the same external search index. Each project needs
#   to have a
2654 # unique id set via EXTERNAL_SEARCH_ID. The search mapping then
#   maps the id of
#   to a relative location where the documentation can be found. The
#   format is:
2656 # EXTRA_SEARCH_MAPPINGS = tagname1=loc1 tagname2=loc2 ...
# This tag requires that the tag SEARCHENGINE is set to YES.
2658 EXTRA_SEARCH_MAPPINGS  =
2660
#
#
# Configuration options related to the LaTeX output
#
#
# If the GENERATELATEX tag is set to YES, doxygen will generate
#   LaTeX output.
2666 # The default value is: YES.
2668 GENERATELATEX           = YES
2670
# The LATEX_OUTPUT tag is used to specify where the LaTeX docs will
#   be put. If a
#   relative path is entered the value of OUTPUT_DIRECTORY will be
#   put in front of
2672 # it.
# The default directory is: latex.
2674 # This tag requires that the tag GENERATELATEX is set to YES.

```

```

2676 LATEX_OUTPUT                = latex
2678 # The LATEX_CMD_NAME tag can be used to specify the LaTeX command
      # name to be
      # invoked.
2680 #
      # Note that when enabling USE_PDFLATEX this option is only used for
      # generating
2682 # bitmaps for formulas in the HTML output, but not in the Makefile
      # that is
      # written to the output directory.
2684 # The default file is: latex.
      # This tag requires that the tag GENERATE_LATEX is set to YES.
2686
2688 LATEX_CMD_NAME                = latex
2688 # The MAKEINDEX_CMD_NAME tag can be used to specify the command
      # name to generate
2690 # index for LaTeX.
      # The default file is: makeindex.
2692 # This tag requires that the tag GENERATE_LATEX is set to YES.
2694
2696 MAKEINDEX_CMD_NAME            = makeindex
2696 # If the COMPACT_LATEX tag is set to YES, doxygen generates more
      # compact LaTeX
      # documents. This may be useful for small projects and may help to
      # save some
2698 # trees in general.
      # The default value is: NO.
2700 # This tag requires that the tag GENERATE_LATEX is set to YES.
2702
2704 COMPACT_LATEX                  = NO
2704 # The PAPER_TYPE tag can be used to set the paper type that is used
      # by the
      # printer.
2706 # Possible values are: a4 (210 x 297 mm), letter (8.5 x 11 inches),
      # legal (8.5 x
      # 14 inches) and executive (7.25 x 10.5 inches).
2708 # The default value is: a4.
      # This tag requires that the tag GENERATE_LATEX is set to YES.
2710
2712 PAPER_TYPE                      = a4
2712 # The EXTRA_PACKAGES tag can be used to specify one or more LaTeX
      # package names
2714 # that should be included in the LaTeX output. The package can be
      # specified just
      # by its name or with the correct syntax as to be used with the
      # LaTeX
2716 # \usepackage command. To get the times font for instance you can
      # specify :
      # EXTRA_PACKAGES=times or EXTRA_PACKAGES={times}
2718 # To use the option intlimits with the amsmath package you can
      # specify:
      # EXTRA_PACKAGES=[intlimits]{amsmath}

```



```

2720 # If left blank no extra packages will be included.
2721 # This tag requires that the tag GENERATELATEX is set to YES.
2722
2723 EXTRA_PACKAGES          =
2724
2725 # The LATEX_HEADER tag can be used to specify a personal LaTeX
2726 # header for the
2727 # generated LaTeX document. The header should contain everything
2728 # until the first
2729 # chapter. If it is left blank doxygen will generate a standard
2730 # header. See
2731 # section "Doxygen usage" for information on how to let doxygen
2732 # write the
2733 # default header to a separate file.
2734 #
2735 # Note: Only use a user-defined header if you know what you are
2736 # doing! The
2737 # following commands have a special meaning inside the header:
2738 # $title,
2739 # $datetime, $date, $doxygenversion, $projectname, $projectnumber,
2740 # $projectbrief, $projectlogo. Doxygen will replace $title with the
2741 # empty
2742 # string, for the replacement values of the other commands the user
2743 # is referred
2744 # to HTML_HEADER.
2745 # This tag requires that the tag GENERATELATEX is set to YES.
2746
2747 LATEX_HEADER             =
2748
2749 # The LATEX_FOOTER tag can be used to specify a personal LaTeX
2750 # footer for the
2751 # generated LaTeX document. The footer should contain everything
2752 # after the last
2753 # chapter. If it is left blank doxygen will generate a standard
2754 # footer. See
2755 # LATEX_HEADER for more information on how to generate a default
2756 # footer and what
2757 # special commands can be used inside the footer.
2758 #
2759 # Note: Only use a user-defined footer if you know what you are
2760 # doing!
2761 # This tag requires that the tag GENERATELATEX is set to YES.
2762
2763 LATEX_FOOTER             =
2764
2765 # The LATEX_EXTRA_STYLESHEET tag can be used to specify additional
2766 # user-defined
2767 # LaTeX style sheets that are included after the standard style
2768 # sheets created
2769 # by doxygen. Using this option one can overrule certain style
2770 # aspects. Doxygen
2771 # will copy the style sheet files to the output directory.
2772 # Note: The order of the extra style sheet files is of importance (
2773 # e.g. the last
2774 # style sheet in the list overrules the setting of the previous
2775 # ones in the
2776 # list).

```

```

2760 # This tag requires that the tag GENERATELATEX is set to YES.
2762 LATEX.EXTRA.STYLESHEET =
2764 # The LATEX.EXTRA.FILES tag can be used to specify one or more
      extra images or
2764 # other source files which should be copied to the LATEX.OUTPUT
      output
      # directory. Note that the files will be copied as-is; there are no
      commands or
2766 # markers available.
      # This tag requires that the tag GENERATELATEX is set to YES.
2768 LATEX.EXTRA.FILES      =
2770 # If the PDF.HYPERLINKS tag is set to YES, the LaTeX that is
      generated is
2772 # prepared for conversion to PDF (using ps2pdf or pdflatex). The
      PDF file will
      # contain links (just like the HTML output) instead of page
      references. This
2774 # makes the output suitable for online browsing using a PDF viewer.
      # The default value is: YES.
2776 # This tag requires that the tag GENERATELATEX is set to YES.

2778 PDF.HYPERLINKS          = YES
2780 # If the USE.PDFLATEX tag is set to YES, doxygen will use pdflatex
      to generate
      # the PDF file directly from the LaTeX files. Set this option to
      YES, to get a
2782 # higher quality PDF documentation.
      # The default value is: YES.
2784 # This tag requires that the tag GENERATELATEX is set to YES.

2786 USE.PDFLATEX            = YES
2788 # If the LATEX.BATCHMODE tag is set to YES, doxygen will add the \
      batchmode
      # command to the generated LaTeX files. This will instruct LaTeX to
      keep running
2790 # if errors occur, instead of asking the user for help. This option
      is also used
      # when generating formulas in HTML.
2792 # The default value is: NO.
      # This tag requires that the tag GENERATELATEX is set to YES.
2794 LATEX.BATCHMODE          = NO
2796 # If the LATEX.HIDE_INDICES tag is set to YES then doxygen will not
      include the
2798 # index chapters (such as File Index, Compound Index, etc.) in the
      output.
      # The default value is: NO.
2800 # This tag requires that the tag GENERATELATEX is set to YES.

2802 LATEX.HIDE_INDICES       = NO

```

```

2804 # If the LATEX_SOURCE_CODE tag is set to YES then doxygen will
      # include source
      # code with syntax highlighting in the LaTeX output.
2806 #
      # Note that which sources are shown also depends on other settings
      # such as
2808 # SOURCE_BROWSER.
      # The default value is: NO.
2810 # This tag requires that the tag GENERATE_LATEX is set to YES.

2812 LATEX_SOURCE_CODE      = NO

2814 # The LATEX_BIB_STYLE tag can be used to specify the style to use
      # for the
      # bibliography, e.g. plainnat, or ieetr. See
2816 # http://en.wikipedia.org/wiki/BibTeX and \cite for more info.
      # The default value is: plain.
2818 # This tag requires that the tag GENERATE_LATEX is set to YES.

2820 LATEX_BIB_STYLE        = plain

2822 # If the LATEX_TIMESTAMP tag is set to YES then the footer of each
      # generated
      # page will contain the date and time when the page was generated.
      # Setting this
2824 # to NO can help when comparing the output of multiple runs.
      # The default value is: NO.
2826 # This tag requires that the tag GENERATE_LATEX is set to YES.

2828 LATEX_TIMESTAMP         = NO

2830 #
      _____

      # Configuration options related to the RTF output
2832 #
      _____

2834 # If the GENERATE_RTF tag is set to YES, doxygen will generate RTF
      # output. The
      # RTF output is optimized for Word 97 and may not look too pretty
      # with other RTF
2836 # readers/editors.
      # The default value is: NO.
2838

2840 GENERATE_RTF            = NO

2842 # The RTF_OUTPUT tag is used to specify where the RTF docs will be
      # put. If a
      # relative path is entered the value of OUTPUT_DIRECTORY will be
      # put in front of
      # it.
2844 # The default directory is: rtf.
      # This tag requires that the tag GENERATE_RTF is set to YES.
2846

```

```

2848 RTF.OUTPUT                = rtf
2849 # If the COMPACTRTF tag is set to YES, doxygen generates more
2850 # compact RTF
2851 # documents. This may be useful for small projects and may help to
2852 # save some
2853 # trees in general.
2854 # The default value is: NO.
2855 # This tag requires that the tag GENERATERTF is set to YES.
2856
2857 COMPACTRTF                = NO
2858 # If the RTF_HYPERLINKS tag is set to YES, the RTF that is
2859 # generated will
2860 # contain hyperlink fields. The RTF file will contain links (just
2861 # like the HTML
2862 # output) instead of page references. This makes the output
2863 # suitable for online
2864 # browsing using Word or some other Word compatible readers that
2865 # support those
2866 # fields.
2867 #
2868 # Note: WordPad (write) and others do not support links.
2869 # The default value is: NO.
2870 # This tag requires that the tag GENERATERTF is set to YES.
2871
2872 RTF_HYPERLINKS            = NO
2873 # Load stylesheet definitions from file. Syntax is similar to
2874 # doxygen's config
2875 # file, i.e. a series of assignments. You only have to provide
2876 # replacements,
2877 # missing definitions are set to their default value.
2878 #
2879 # See also section "Doxygen usage" for information on how to
2880 # generate the
2881 # default style sheet that doxygen normally uses.
2882 # This tag requires that the tag GENERATERTF is set to YES.
2883
2884 RTF_STYLESHEET_FILE       =
2885 # Set optional variables used in the generation of an RTF document.
2886 # Syntax is
2887 # similar to doxygen's config file. A template extensions file can
2888 # be generated
2889 # using doxygen -e rtf extensionFile.
2890 # This tag requires that the tag GENERATERTF is set to YES.
2891
2892 RTF_EXTENSIONS_FILE       =
2893 # If the RTF_SOURCE_CODE tag is set to YES then doxygen will
2894 # include source code
2895 # with syntax highlighting in the RTF output.
2896 #
2897 # Note that which sources are shown also depends on other settings
2898 # such as
2899 # SOURCE_BROWSER.

```

```

# The default value is: NO.
2892 # This tag requires that the tag GENERATERTF is set to YES.

2894 RTF.SOURCE.CODE          = NO

2896 #
# Configuration options related to the man page output
2898 #

# If the GENERATEMAN tag is set to YES, doxygen will generate man
# pages for
# classes and files.
2900 # The default value is: NO.
2902 # The default value is: NO.

2904 GENERATEMAN              = NO

2906 # The MAN.OUTPUT tag is used to specify where the man pages will be
# put. If a
# relative path is entered the value of OUTPUT.DIRECTORY will be
# put in front of
2908 # it. A directory man3 will be created inside the directory
# specified by
# MAN.OUTPUT.
2910 # The default directory is: man.
# This tag requires that the tag GENERATEMAN is set to YES.
2912
2914 MAN.OUTPUT                = man

# The MAN.EXTENSION tag determines the extension that is added to
# the generated
2916 # man pages. In case the manual section does not start with a
# number, the number
# 3 is prepended. The dot (.) at the beginning of the MAN.EXTENSION
# tag is
2918 # optional.
# The default value is: .3.
2920 # This tag requires that the tag GENERATEMAN is set to YES.

2922 MAN.EXTENSION             = .3

2924 # The MAN.SUBDIR tag determines the name of the directory created
# within
# MAN.OUTPUT in which the man pages are placed. If defaults to man
# followed by
2926 # MAN.EXTENSION with the initial . removed.
# This tag requires that the tag GENERATEMAN is set to YES.
2928
2930 MAN.SUBDIR                 =

# If the MAN.LINKS tag is set to YES and doxygen generates man
# output, then it
2932 # will generate one additional man file for each entity documented
# in the real

```

```

# man page(s). These additional files only source the real man page
# , but without
2934 # them the man command would be unable to find the correct page.
# The default value is: NO.
2936 # This tag requires that the tag GENERATEMAN is set to YES.

2938 MAN_LINKS                = NO

2940 #
#
# Configuration options related to the XML output
2942 #
#
# If the GENERATEXML tag is set to YES, doxygen will generate an
# XML file that
# captures the structure of the code including all documentation.
2946 # The default value is: NO.

2948 GENERATEXML              = NO

2950 # The XMLOUTPUT tag is used to specify where the XML pages will be
# put. If a
# relative path is entered the value of OUTPUT_DIRECTORY will be
# put in front of
2952 # it.
# The default directory is: xml.
2954 # This tag requires that the tag GENERATEXML is set to YES.

2956 XMLOUTPUT                = xml

2958 # If the XMLPROGRAMLISTING tag is set to YES, doxygen will dump
# the program
# listings (including syntax highlighting and cross-referencing
# information) to
2960 # the XML output. Note that enabling this will significantly
# increase the size
# of the XML output.
2962 # The default value is: YES.
# This tag requires that the tag GENERATEXML is set to YES.

2964 XMLPROGRAMLISTING        = YES

2966 #
#
# Configuration options related to the DOCBOOK output
2968 #
#
# If the GENERATEDOCBOOK tag is set to YES, doxygen will generate
# Docbook files
2970 # that can be used to generate PDF.
2972 # The default value is: NO.

```

```

2974 GENERATE_DOCBOOK      = NO
2976
2977 # The DOCBUILD_OUTPUT tag is used to specify where the Docbook pages
2978 # will be put.
2979 # If a relative path is entered the value of OUTPUT_DIRECTORY will
2980 # be put in
2981 # front of it.
2982 # The default directory is: docbook.
2983 # This tag requires that the tag GENERATE_DOCBOOK is set to YES.
2984
2985 DOCBUILD_OUTPUT        = docbook
2986
2987 # If the DOCBUILD_PROGRAMLISTING tag is set to YES, doxygen will
2988 # include the
2989 # program listings (including syntax highlighting and cross-
2990 # referencing
2991 # information) to the DOCBUILD output. Note that enabling this will
2992 # significantly
2993 # increase the size of the DOCBUILD output.
2994 # The default value is: NO.
2995 # This tag requires that the tag GENERATE_DOCBOOK is set to YES.
2996
2997 DOCBUILD_PROGRAMLISTING = NO
2998
2999 #
3000
3001 # Configuration options for the AutoGen Definitions output
3002 #
3003
3004 # If the GENERATE_AUTOGEN_DEF tag is set to YES, doxygen will
3005 # generate an
3006 # AutoGen Definitions (see http://autogen.sf.net) file that
3007 # captures the
3008 # structure of the code including all documentation. Note that this
3009 # feature is
3010 # still experimental and incomplete at the moment.
3011 # The default value is: NO.
3012
3013 GENERATE_AUTOGEN_DEF    = NO
3014
3015 #
3016
3017 # Configuration options related to the Perl module output
3018 #
3019
3020 # If the GENERATE_PERLMOD tag is set to YES, doxygen will generate
3021 # a Perl module
3022 # file that captures the structure of the code including all
3023 # documentation.
3024 #
3025

```

```

# Note that this feature is still experimental and incomplete at
# the moment.
3014 # The default value is: NO.

GENERATE_PERLMOD      = NO

3018 # If the PERLMODLATEX tag is set to YES, doxygen will generate the
# necessary
# Makefile rules, Perl scripts and LaTeX code to be able to
# generate PDF and DVI
3020 # output from the Perl module output.
# The default value is: NO.
3022 # This tag requires that the tag GENERATE_PERLMOD is set to YES.

PERLMODLATEX          = NO

3026 # If the PERLMODPRETTY tag is set to YES, the Perl module output
# will be nicely
# formatted so it can be parsed by a human reader. This is useful
# if you want to
3028 # understand what is going on. On the other hand, if this tag is
# set to NO, the
# size of the Perl module output will be much smaller and Perl will
# parse it
3030 # just the same.
# The default value is: YES.
3032 # This tag requires that the tag GENERATE_PERLMOD is set to YES.

PERLMODPRETTY          = YES

3036 # The names of the make variables in the generated doxyrules.make
# file are
# prefixed with the string contained in PERLMOD_MAKEVAR_PREFIX.
# This is useful
3038 # so different doxyrules.make files included by the same Makefile
# don't
# overwrite each other's variables.
3040 # This tag requires that the tag GENERATE_PERLMOD is set to YES.

PERLMOD_MAKEVAR_PREFIX =

3044 #
# _____

# Configuration options related to the preprocessor
3046 #
# _____

3048 # If the ENABLE_PREPROCESSING tag is set to YES, doxygen will
# evaluate all
# C-preprocessor directives found in the sources and include files.
3050 # The default value is: YES.

ENABLE_PREPROCESSING   = YES

3054 # If the MACRO_EXPANSION tag is set to YES, doxygen will expand all

```



```

macro names
# in the source code. If set to NO, only conditional compilation
  will be
3056 # performed. Macro expansion can be done in a controlled way by
      setting
# EXPAND_ONLY_PREDEF to YES.
3058 # The default value is: NO.
# This tag requires that the tag ENABLE_PREPROCESSING is set to YES
.
3060 MACRO_EXPANSION          = NO
3062 # If the EXPAND_ONLY_PREDEF and MACRO_EXPANSION tags are both set
  to YES then
3064 # the macro expansion is limited to the macros specified with the
  PREDEFINED and
# EXPAND_AS_DEFINED tags.
3066 # The default value is: NO.
# This tag requires that the tag ENABLE_PREPROCESSING is set to YES
.
3068 EXPAND_ONLY_PREDEF      = NO
3070 # If the SEARCH_INCLUDES tag is set to YES, the include files in
  the
3072 # INCLUDE_PATH will be searched if a #include is found.
# The default value is: YES.
3074 # This tag requires that the tag ENABLE_PREPROCESSING is set to YES
.
3076 SEARCH_INCLUDES        = YES
3078 # The INCLUDE_PATH tag can be used to specify one or more
  directories that
# contain include files that are not input files but should be
  processed by the
3080 # preprocessor.
# This tag requires that the tag SEARCH_INCLUDES is set to YES.
3082 INCLUDE_PATH            =
3084 # You can use the INCLUDE_FILE_PATTERNS tag to specify one or more
  wildcard
3086 # patterns (like *.h and *.hpp) to filter out the header-files in
  the
# directories. If left blank, the patterns specified with
  FILE_PATTERNS will be
3088 # used.
# This tag requires that the tag ENABLE_PREPROCESSING is set to YES
.
3090 INCLUDE_FILE_PATTERNS   =
3092 # The PREDEFINED tag can be used to specify one or more macro names
  that are
3094 # defined before the preprocessor is started (similar to the -D
  option of e.g.

```

```

# gcc). The argument of the tag is a list of macros of the form:
# name or
3096 # name=definition (no spaces). If the definition and the "=" are
# omitted, "=1"
# is assumed. To prevent a macro definition from being undefined
# via #undef or
3098 # recursively expanded use the := operator instead of the =
# operator.
# This tag requires that the tag ENABLEPREPROCESSING is set to YES
.

3100 PREDEFINED =
3102 # If the MACRO_EXPANSION and EXPAND_ONLY_PREDEF tags are set to YES
# then this
3104 # tag can be used to specify a list of macro names that should be
# expanded. The
# macro definition that is found in the sources will be used. Use
# the PREDEFINED
3106 # tag if you want to use a different macro definition that
# overrules the
# definition found in the source code.
3108 # This tag requires that the tag ENABLEPREPROCESSING is set to YES
.

3110 EXPAND_AS_DEFINED =

3112 # If the SKIP_FUNCTION_MACROS tag is set to YES then doxygen's
# preprocessor will
# remove all references to function-like macros that are alone on a
# line, have
3114 # an all uppercase name, and do not end with a semicolon. Such
# function macros
# are typically used for boiler-plate code, and will confuse the
# parser if not
3116 # removed.
# The default value is: YES.
3118 # This tag requires that the tag ENABLEPREPROCESSING is set to YES
.

3120 SKIP_FUNCTION_MACROS = YES

3122 #


---


# Configuration options related to external references
3124 #


---



3126 # The TAGFILES tag can be used to specify one or more tag files.
# For each tag
# file the location of the external documentation should be added.
# The format of
3128 # a tag file without this location is as follows:
# TAGFILES = file1 file2 ...
3130 # Adding location for the tag files is done as follows:

```

```

# TAGFILES = file1=loc1 "file2 = loc2" ...
3132 # where loc1 and loc2 can be relative or absolute paths or URLs.
# See the
# section "Linking to external documentation" for more information
# about the use
3134 # of tag files.
# Note: Each tag file must have a unique name (where the name does
# NOT include
3136 # the path). If a tag file is not located in the directory in which
# doxygen is
# run, you must also specify the path to the tagfile here.
3138 TAGFILES =
3140 # When a file name is specified after GENERATE_TAGFILE, doxygen
# will create a
3142 # tag file that is based on the input files it reads. See section "
# Linking to
# external documentation" for more information about the usage of
# tag files.
3144 GENERATE_TAGFILE =
3146 # If the ALLEXTERNALS tag is set to YES, all external class will be
# listed in
3148 # the class index. If set to NO, only the inherited external
# classes will be
# listed.
3150 # The default value is: NO.
3152 ALLEXTERNALS = NO
3154 # If the EXTERNAL_GROUPS tag is set to YES, all external groups
# will be listed
# in the modules index. If set to NO, only the current project's
# groups will be
3156 # listed.
# The default value is: YES.
3158 EXTERNAL_GROUPS = YES
3160 # If the EXTERNAL_PAGES tag is set to YES, all external pages will
# be listed in
3162 # the related pages index. If set to NO, only the current project's
# pages will
# be listed.
3164 # The default value is: YES.
3166 EXTERNAL_PAGES = YES
3168 # The PERL_PATH should be the absolute path and name of the perl
# script
# interpreter (i.e. the result of 'which perl').
3170 # The default file (with absolute path) is: /usr/bin/perl.
3172 PERL_PATH = /usr/bin/perl

```

```

3174 |
3176 | #
3178 | # Configuration options related to the dot tool
3180 | # If the CLASS_DIAGRAMS tag is set to YES, doxygen will generate a
3182 | # class diagram
3184 | # (in HTML and LaTeX) for classes with base or super classes.
3186 | # Setting the tag to
3188 | # NO turns the diagrams off. Note that this option also works with
3190 | # HAVE_DOT
3192 | # disabled, but it is recommended to install and use dot, since it
3194 | # yields more
3196 | # powerful graphs.
3198 | # The default value is: YES.
3200 | CLASS_DIAGRAMS = YES
3202 | # You can define message sequence charts within doxygen comments
3204 | # using the \msc
3206 | # command. Doxygen will then run the mscgen tool (see:
3208 | # http://www.mcternan.me.uk/mscgen/) to produce the chart and
3210 | # insert it in the
3212 | # documentation. The MSCGEN_PATH tag allows you to specify the
3214 | # directory where
3216 | # the mscgen tool resides. If left empty the tool is assumed to be
3218 | # found in the
3220 | # default search path.
3222 | MSCGEN_PATH =
3224 | # You can include diagrams made with dia in doxygen documentation.
3226 | # Doxygen will
3228 | # then run dia to produce the diagram and insert it in the
3230 | # documentation. The
3232 | # DIA_PATH tag allows you to specify the directory where the dia
3234 | # binary resides.
3236 | # If left empty dia is assumed to be found in the default search
3238 | # path.
3240 | DIA_PATH =
3242 | # If set to YES the inheritance and collaboration graphs will hide
3244 | # inheritance
3246 | # and usage relations if the target is undocumented or is not a
3248 | # class.
3250 | # The default value is: YES.
3252 | HIDE_UNDOC_RELATIONS = YES
3254 | # If you set the HAVE_DOT tag to YES then doxygen will assume the
3256 | # dot tool is

```

```

3212 # available from the path. This tool is part of Graphviz (see:
3213 # http://www.graphviz.org/), a graph visualization toolkit from AT&
3214 # Bell Labs. The other options in this section have no effect if
3215 # this option is
3216 # set to NO
3217 # The default value is: YES.
3218 HAVE_DOT = YES
3219
3220 # The DOT_NUM_THREADS specifies the number of dot invocations
3221 # doxygen is allowed
3222 # to run in parallel. When set to 0 doxygen will base this on the
3223 # number of
3224 # processors available in the system. You can set it explicitly to
3225 # a value
3226 # larger than 0 to get control over the balance between CPU load
3227 # and processing
3228 # speed.
3229 # Minimum value: 0, maximum value: 32, default value: 0.
3230 # This tag requires that the tag HAVE_DOT is set to YES.
3231
3232 DOT_NUM_THREADS = 0
3233
3234 # When you want a differently looking font in the dot files that
3235 # doxygen
3236 # generates you can specify the font name using DOT_FONTNAME. You
3237 # need to make
3238 # sure dot is able to find the font, which can be done by putting
3239 # it in a
3240 # standard location or by setting the DOT_FONTPATH environment
3241 # variable or by
3242 # setting DOT_FONTPATH to the directory containing the font.
3243 # The default value is: Helvetica.
3244 # This tag requires that the tag HAVE_DOT is set to YES.
3245
3246 DOT_FONTNAME = Helvetica
3247
3248 # The DOT_FONTSIZE tag can be used to set the size (in points) of
3249 # the font of
3250 # dot graphs.
3251 # Minimum value: 4, maximum value: 24, default value: 10.
3252 # This tag requires that the tag HAVE_DOT is set to YES.
3253
3254 DOT_FONTSIZE = 10
3255
3256 # By default doxygen will tell dot to use the default font as
3257 # specified with
3258 # DOT_FONTNAME. If you specify a different font using DOT_FONTNAME
3259 # you can set
3260 # the path where dot can find it using this tag.
3261 # This tag requires that the tag HAVE_DOT is set to YES.
3262
3263 DOT_FONTPATH =
3264
3265 # If the CLASS_GRAPH tag is set to YES then doxygen will generate a
3266 # graph for

```

```

# each documented class showing the direct and indirect inheritance
# relations.
3256 # Setting this tag to YES will force the CLASS_DIAGRAMS tag to NO.
# The default value is: YES.
3258 # This tag requires that the tag HAVE_DOT is set to YES.

3260 CLASS_GRAPH          = YES

3262 # If the COLLABORATION_GRAPH tag is set to YES then doxygen will
# generate a
# graph for each documented class showing the direct and indirect
# implementation
3264 # dependencies (inheritance, containment, and class references
# variables) of the
# class with other documented classes.
3266 # The default value is: YES.
# This tag requires that the tag HAVE_DOT is set to YES.
3268
COLLABORATION_GRAPH    = YES

3270 # If the GROUP_GRAPHS tag is set to YES then doxygen will generate
# a graph for
3272 # groups, showing the direct groups dependencies.
# The default value is: YES.
3274 # This tag requires that the tag HAVE_DOT is set to YES.

3276 GROUP_GRAPHS         = YES

3278 # If the UML_LOOK tag is set to YES, doxygen will generate
# inheritance and
# collaboration diagrams in a style similar to the OMG's Unified
# Modeling
3280 # Language.
# The default value is: NO.
3282 # This tag requires that the tag HAVE_DOT is set to YES.

3284 UML_LOOK              = YES

3286 # If the UML_LOOK tag is enabled, the fields and methods are shown
# inside the
# class node. If there are many fields or methods and many nodes
# the graph may
3288 # become too big to be useful. The UML_LIMIT_NUM_FIELDS threshold
# limits the
# number of items for each type to make the size more manageable.
# Set this to 0
3290 # for no limit. Note that the threshold may be exceeded by 50%
# before the limit
# is enforced. So when you set the threshold to 10, up to 15 fields
# may appear,
3292 # but if the number exceeds 15, the total amount of fields shown is
# limited to
# 10.
3294 # Minimum value: 0, maximum value: 100, default value: 10.
# This tag requires that the tag HAVE_DOT is set to YES.
3296
UML_LIMIT_NUM_FIELDS   = 10

```

```

3298 # If the TEMPLATE_RELATIONS tag is set to YES then the inheritance
      and
3300 # collaboration graphs will show the relations between templates
      and their
      # instances.
3302 # The default value is: NO.
      # This tag requires that the tag HAVE_DOT is set to YES.
3304
      TEMPLATE_RELATIONS          = NO
3306
      # If the INCLUDE_GRAPH, ENABLE_PREPROCESSING and SEARCH_INCLUDES
      tags are set to
3308 # YES then doxygen will generate a graph for each documented file
      showing the
      # direct and indirect include dependencies of the file with other
      documented
3310 # files.
      # The default value is: YES.
3312 # This tag requires that the tag HAVE_DOT is set to YES.
3314
      INCLUDE_GRAPH              = YES
3316
      # If the INCLUDED_BY_GRAPH, ENABLE_PREPROCESSING and
      SEARCH_INCLUDES tags are
      # set to YES then doxygen will generate a graph for each documented
      file showing
3318 # the direct and indirect include dependencies of the file with
      other documented
      # files.
3320 # The default value is: YES.
      # This tag requires that the tag HAVE_DOT is set to YES.
3322
      INCLUDED_BY_GRAPH          = YES
3324
      # If the CALL_GRAPH tag is set to YES then doxygen will generate a
      call
3326 # dependency graph for every global function or class method.
      #
3328 # Note that enabling this option will significantly increase the
      time of a run.
      # So in most cases it will be better to enable call graphs for
      selected
3330 # functions only using the \callgraph command. Disabling a call
      graph can be
      # accomplished by means of the command \hidecallgraph.
3332 # The default value is: NO.
      # This tag requires that the tag HAVE_DOT is set to YES.
3334
      CALL_GRAPH                 = NO
3336
      # If the CALLER_GRAPH tag is set to YES then doxygen will generate
      a caller
3338 # dependency graph for every global function or class method.
      #
3340 # Note that enabling this option will significantly increase the
      time of a run.

```

```

# So in most cases it will be better to enable caller graphs for
# selected
3342 # functions only using the \callergraph command. Disabling a caller
# graph can be
# accomplished by means of the command \hidecallergraph.
3344 # The default value is: NO.
# This tag requires that the tag HAVE_DOT is set to YES.
3346
CALLER_GRAPH          = NO
3348
# If the GRAPHICAL_HIERARCHY tag is set to YES then doxygen will
# graphical
3350 # hierarchy of all classes instead of a textual one.
# The default value is: YES.
3352 # This tag requires that the tag HAVE_DOT is set to YES.

GRAPHICAL_HIERARCHY   = YES
3354

# If the DIRECTORY_GRAPH tag is set to YES then doxygen will show
# the
# dependencies a directory has on other directories in a graphical
# way. The
3358 # dependency relations are determined by the #include relations
# between the
# files in the directories.
3360 # The default value is: YES.
# This tag requires that the tag HAVE_DOT is set to YES.
3362

DIRECTORY_GRAPH       = YES
3364

# The DOT_IMAGE_FORMAT tag can be used to set the image format of
# the images
3366 # generated by dot. For an explanation of the image formats see the
# section
# output formats in the documentation of the dot tool (Graphviz (
# see:
3368 # http://www.graphviz.org/)).
# Note: If you choose svg you need to set HTML_FILE_EXTENSION to
# xhtml in order
3370 # to make the SVG files visible in IE 9+ (other browsers do not
# have this
# requirement).
3372 # Possible values are: png, png:cairo, png:cairo:cairo, png:cairo:
# gd, png:gd,
# png:gd:gd, jpg, jpg:cairo, jpg:cairo:gd, jpg:gd, jpg:gd:gd, gif,
# gif:cairo,
3374 # gif:cairo:gd, gif:gd, gif:gd:gd, svg, png:gd, png:gd:gd, png:
# cairo,
# png:cairo:gd, png:cairo:cairo, png:cairo:gdiplus, png:gdiplus and
3376 # png:gdiplus:gdiplus.
# The default value is: png.
3378 # This tag requires that the tag HAVE_DOT is set to YES.

DOT_IMAGE_FORMAT      = png
3380
3382
# If DOT_IMAGE_FORMAT is set to svg, then this option can be set to

```



```

3384 # enable generation of interactive SVG images that allow zooming
      YES to
      and panning.
3386 #
      # Note that this requires a modern browser other than Internet
      Explorer. Tested
      # and working are Firefox, Chrome, Safari, and Opera.
3388 # Note: For IE 9+ you need to set HTMLFILE_EXTENSION to xhtml in
      order to make
      # the SVG files visible. Older versions of IE do not have SVG
      support.
3390 # The default value is: NO.
      # This tag requires that the tag HAVE_DOT is set to YES.
3392 INTERACTIVE_SVG          = NO
3394 # The DOT_PATH tag can be used to specify the path where the dot
      tool can be
3396 # found. If left blank, it is assumed the dot tool can be found in
      the path.
      # This tag requires that the tag HAVE_DOT is set to YES.
3398 DOT_PATH                  =
3400 # The DOTFILE_DIRS tag can be used to specify one or more
      directories that
3402 # contain dot files that are included in the documentation (see the
      \dotfile
      # command).
3404 # This tag requires that the tag HAVE_DOT is set to YES.
3406 DOTFILE_DIRS              =
3408 # The MSCFILE_DIRS tag can be used to specify one or more
      directories that
      # contain msc files that are included in the documentation (see the
      \mscfile
3410 # command).
3412 MSCFILE_DIRS              =
3414 # The DIAFILE_DIRS tag can be used to specify one or more
      directories that
      # contain dia files that are included in the documentation (see the
      \diafile
3416 # command).
3418 DIAFILE_DIRS              =
3420 # When using plantuml, the PLANTUML_JAR_PATH tag should be used to
      specify the
      # path where java can find the plantuml.jar file. If left blank, it
      is assumed
3422 # PlantUML is not used or called during a preprocessing step.
      Doxygen will
      # generate a warning when it encounters a \startuml command in this
      case and

```

```

3424 # will not generate output for the diagram.
3426 PLANTUML_JAR_PATH      =
3428 # When using plantuml, the PLANTUML_CFG_FILE tag can be used to
    # specify a
    # configuration file for plantuml.
3430 PLANTUML_CFG_FILE      =
3432 # When using plantuml, the specified paths are searched for files
    # specified by
3434 # the !include statement in a plantuml block.
3436 PLANTUML_INCLUDE_PATH  =
3438 # The DOT_GRAPH_MAX_NODES tag can be used to set the maximum number
    # of nodes
    # that will be shown in the graph. If the number of nodes in a
    # graph becomes
3440 # larger than this value, doxygen will truncate the graph, which is
    # visualized
    # by representing a node as a red box. Note that doxygen if the
    # number of direct
3442 # children of the root node in a graph is already larger than
    # DOT_GRAPH_MAX_NODES then the graph will not be shown at all. Also
    # note that
3444 # the size of a graph can be further restricted by
    # MAX_DOT_GRAPH_DEPTH.
    # Minimum value: 0, maximum value: 10000, default value: 50.
3446 # This tag requires that the tag HAVE_DOT is set to YES.
3448 DOT_GRAPH_MAX_NODES    = 50
3450 # The MAX_DOT_GRAPH_DEPTH tag can be used to set the maximum depth
    # of the graphs
    # generated by dot. A depth value of 3 means that only nodes
    # reachable from the
3452 # root by following a path via at most 3 edges will be shown. Nodes
    # that lay
    # further from the root node will be omitted. Note that setting
    # this option to 1
3454 # or 2 may greatly reduce the computation time needed for large
    # code bases. Also
    # note that the size of a graph can be further restricted by
3456 # DOT_GRAPH_MAX_NODES. Using a depth of 0 means no depth
    # restriction.
    # Minimum value: 0, maximum value: 1000, default value: 0.
3458 # This tag requires that the tag HAVE_DOT is set to YES.
3460 MAX_DOT_GRAPH_DEPTH    = 0
3462 # Set the DOT_TRANSPARENT tag to YES to generate images with a
    # transparent
    # background. This is disabled by default, because dot on Windows
    # does not seem
3464 # to support this out of the box.

```

```

#
3466 # Warning: Depending on the platform used, enabling this option may
      # lead to
      # badly anti-aliased labels on the edges of a graph (i.e. they
      # become hard to
3468 # read).
      # The default value is: NO.
3470 # This tag requires that the tag HAVE_DOT is set to YES.

DOT_TRANSPARENT          = NO

3474 # Set the DOT_MULTITARGETS tag to YES to allow dot to generate
      # multiple output
      # files in one run (i.e. multiple -o and -T options on the command
      # line). This
3476 # makes dot run faster, but since only newer versions of dot
      # (>1.8.10) support
      # this, this feature is disabled by default.
3478 # The default value is: NO.
      # This tag requires that the tag HAVE_DOT is set to YES.
3480
DOT_MULTITARGETS          = NO
3482
      # If the GENERATE_LEGEND tag is set to YES doxygen will generate a
      # legend page
3484 # explaining the meaning of the various boxes and arrows in the dot
      # generated
      # graphs.
3486 # The default value is: YES.
      # This tag requires that the tag HAVE_DOT is set to YES.
3488
GENERATE_LEGEND           = YES
3490
      # If the DOT_CLEANUP tag is set to YES, doxygen will remove the
      # intermediate dot
3492 # files that are used to generate the various graphs.
      # The default value is: YES.
3494 # This tag requires that the tag HAVE_DOT is set to YES.

DOT_CLEANUP               = YES
3496

```

codes/Doxygen_config_files/Doxyfile_Build