Direct HZD Optimization

1.0

Generated by Doxygen 1.8.12

Contents

1 Contributed Developers

1.1 Ayonga Hereid

http://www.bipedalrobotics.com

- Optimization & Control
- Mathematica Modeling
- Matlab Simulation

2 Changes in Direct HZD Optimization Version 1.2

Class MatlabDocMaker

(dw, 2011-11-27)

- Included documentation creation for the Windows platform and combined the old methods into one (small effective differences)
- No longer storing the doxygen binary file in the prefs as a lot of tools must be present on the path anyways. The new paradigm is to expect all required 3rd-party programmes to be available on PATH. As backup the configuration files directory is added to the Matlab PATH environment **nonpermanently** and any executables found there will thus also be usable.
- Included checks for `dot` and `latex` at the setup stage to recommend installation of those tools if not present (the default doxygen settings in Doxyfile.m4 are to use both)

(dw, 2011-11-08) Improved the createUnix method by displaying the warnings and writing the output to a log file afterwards. Not using cprintf anymore as this is 3rd party software.

(dw, 2011-11-07) Fixed a recursion bug caused by copy and paste. Now the preferences are stored on an per-application basis.

(dw, 2011-11-04) Changed the name to MatlabDocMaker in order to export it into the mtoc++ distribution later.

3 Changes in Direct HZD Optimization Version 1.3

Class MatlabDocMaker

(dw, 2012-02-16)

- Now also collecting error messages from mtocpp_post and adding them to the warnings.log file.
- Added the directive "LD_LIBRARY_PATH= " for unix systems, as MatLab sets it inside its executing environment. This can lead to errors if doxygen and/or mtoc++ have been built using never GLIBC (libstd) versions than the one shipped with MatLab.

(dw, 2012-01-16)

- Properly using the correct file separators everywhere now
- Hyperlinked the log file so it can be opened directly

(dw, 2012-01-14) Not displaying the "generated warnings"-text if there have not been any during documentation creation.

4 Changes in Direct HZD Optimization Version 1.4

Class MatlabDocMaker

(dw, 2012-10-18) Removed `m4` dependency and included constant properties for configuration file names.

(dw, 2012-09-27) Added automatic dot Graphviz tool detection on call to create.

5 Changes in Direct HZD Optimization Version 1.5

Class MatlabDocMaker

(dw, 2013-12-03) Fixed default value selection for properties, now not having set a description or logo does not cause an error to be thrown.

(dw, 2013-02-21) Fixed the callback for suggested direct documentation creation after MatlabDocMaker.setup (Thanks to Aurelien Queffurust)

(dw, 2013-02-12) Also added the escaping for the Logo file. Thanks to Chris Olien for the hint.

(dw, 2013-01-07) Included some backslash escaping for paths on windows platforms. Thanks to MathWorks Pilot Engineer '`Arvind Jayaraman`' for providing the feedback and code!

6 New features in Direct HZD Optimization Version 1.2

Class MatlabDocMaker

(dw, 2011-10-13) Added this class and moved documentation related stuff here from the KerMor class.

7 New features in Direct HZD Optimization Version 1.4

Class MatlabDocMaker

(dw, 2012-10-16)

- Added two more configuration variables "ProjectDescription" and "ProjectLogo" for easier configuration of the MatlabDocMaker in many cases. Thanks to Wolfgang Mennerich `http://www.mathworks.↩com/matlabcentral/fileexchange/authors/272859` for the suggestion.
- Restructured the configuration, now only the project name function has to be implemented (the preferences tag depends on it, there might be more than one project within the same Matlab installation whos documentation is created using this tool). The rest can be provided either at setup time or later via suitable setters for the version, description and logo.
- Automatically setting HaveDot in the doxygen config whenever its found on the environment path.
- Added basic support for LaTeX documentation creation. Using the parameter `latex`=true for the create method creates the LaTeX version of the documentation in a folder "latex" in the OutputDirectory (default behaviour)

8 Hierarchical Index

8.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

9 Class Index

9.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

10 File Index

10.1 File List

Here is a list of all files with brief descriptions:

11 Class Documentation

11.1 cell Class Reference

A MatLab cell array or matrix.

11.1.1 Detailed Description

A MatLab cell array or matrix.

This class is an artificially created class in doxygen to allow more precise type declarations

The documentation for this class was generated from the following file:

- /home/ayonga/Dropbox/research/dzopt/direct_hzd_optimization/docs/config/class_substitutes.c

11.2 char Class Reference

A MatLab character array.

11.2.1 Detailed Description

A MatLab character array.

This class is an artificially created class in doxygen to allow more precise type declarations and represents string-like types.

The documentation for this class was generated from the following file:

- /home/ayonga/Dropbox/research/dzopt/direct_hzd_optimization/docs/config/class_substitutes.c

11.3 colvec Class Reference

A matlab column vector.

11.3.1 Detailed Description

A matlab column vector.

This class is an artificially created class in doxygen to allow more precise type declarations

The documentation for this class was generated from the following file:

- /home/ayonga/Dropbox/research/dzopt/direct_hzd_optimization/docs/config/class_substitutes.c

11.4 double Class Reference

A double value.

11.4.1 Detailed Description

A double value.

This class is an artificially created class in doxygen to allow more precise type declarations. The MatLab type associated with this class is double.

The documentation for this class was generated from the following file:

- /home/ayonga/Dropbox/research/dzopt/direct_hzd_optimization/docs/config/class_substitutes.c

11.5 function_handle Class Reference

A MatLab function handle.

11.5.1 Detailed Description

A MatLab function handle.

This class is an artificially created class in doxygen to allow more precise type declarations

The documentation for this class was generated from the following file:

- /home/ayonga/Dropbox/research/dzopt/direct_hzd_optimization/docs/config/class_substitutes.c

11.6 handle Class Reference

Matlab's base handle class (documentation generation substitute)

Public Attributes

- matlabtypesubstitute addlistener

  Creates a listener for the specified event and assigns a callback function to execute when the event occurs.
- matlabtypesubstitute notify

  Broadcast a notice that a specific event is occurring on a specified handle object or array of handle objects.
- matlabtypesubstitute delete

  Handle object destructor method that is called when the object's lifecycle ends.
- matlabtypesubstitute disp

  Handle object disp method which is called by the display method. See the MATLAB disp function.
- matlabtypesubstitute display

  Handle object display method called when MATLAB software interprets an expression returning a handle object that is not terminated by a semicolon. See the MATLAB display function.
- matlabtypesubstitute findobj

  Finds objects matching the specified conditions from the input array of handle objects.
- matlabtypesubstitute findprop

  Returns a meta.property objects associated with the specified property name.
- matlabtypesubstitute fields

  Returns a cell array of string containing the names of public properties.
- matlabtypesubstitute fieldnames

  Returns a cell array of string containing the names of public properties. See the MATLAB fieldnames function.
- matlabtypesubstitute isvalid

  Returns a logical array in which elements are true if the corresponding elements in the input array are valid handles. This method is Sealed so you cannot override it in a handle subclass.
- matlabtypesubstitute eq

  Relational functions example. See details for more information.
- matlabtypesubstitute transpose

  Transposes the elements of the handle object array.
- matlabtypesubstitute permute

  Rearranges the dimensions of the handle object array. See the MATLAB permute function.
- matlabtypesubstitute reshape

  hanges the dimensions of the handle object array to the specified dimensions. See the MATLAB reshape function.
- matlabtypesubstitute sort

  ort the handle objects in any array in ascending or descending order.

11.6.1 Detailed Description

Matlab's base handle class (documentation generation substitute)

As doxygen does not know the class "handle" from itself, many classes do not get rendered within the documentation and the correct root class is not even displayed. This workaround guarantees a correct (also graphical) representation of the class hierarchy.

Note here that by having the type handle it could also mean to have a vector or matrix of handles.

Definition at line 91 of file class_substitutes.c.

11.6.2 Member Data Documentation

11.6.2.1 matlabtypesubstitute handle::addlistener

Creates a listener for the specified event and assigns a callback function to execute when the event occurs.

See also

notify

Definition at line 106 of file class_substitutes.c.

11.6.2.2 matlabtypesubstitute handle::delete

Handle object destructor method that is called when the object's lifecycle ends.

Definition at line 117 of file class_substitutes.c.

11.6.2.3 matlabtypesubstitute handle::disp

Handle object disp method which is called by the display method. See the MATLAB disp function.

Definition at line 122 of file class_substitutes.c.

11.6.2.4 matlabtypesubstitute handle::display

Handle object display method called when MATLAB software interprets an expression returning a handle object that is not terminated by a semicolon. See the MATLAB display function.

Definition at line 127 of file class_substitutes.c.

11.6.2.5 matlabtypesubstitute handle::eq

Relational functions example. See details for more information.

Other possible relational operators:

-ne -lt -le -gt -ge

Relational functions return a logical array of the same size as the pair of input handle object arrays. Comparisons use a number associated with each handle. You can assume that the same two handles will compare as equal and the repeated comparison of any two handles will yield the same result in the same MATLAB session. Different handles are always not-equal. The order of handles is purely arbitrary, but consistent.

Definition at line 167 of file class_substitutes.c.

11.6.2.6 matlabtypesubstitute handle::fieldnames

Returns a cell array of string containing the names of public properties. See the MATLAB fieldnames function.

Definition at line 147 of file class_substitutes.c.

11.6.2.7 matlabtypesubstitute handle::fields

Returns a cell array of string containing the names of public properties.

Definition at line 142 of file class_substitutes.c.

11.6.2.8 matlabtypesubstitute handle::findobj

Finds objects matching the specified conditions from the input array of handle objects.

Definition at line 132 of file class_substitutes.c.

11.6.2.9 matlabtypesubstitute handle::findprop

Returns a meta.property objects associated with the specified property name.

Definition at line 137 of file class_substitutes.c.

11.6.2.10 matlabtypesubstitute handle::isvalid

Returns a logical array in which elements are true if the corresponding elements in the input array are valid handles. This method is Sealed so you cannot override it in a handle subclass.

Definition at line 153 of file class_substitutes.c.

11.6.2.11 matlabtypesubstitute handle::notify

Broadcast a notice that a specific event is occurring on a specified handle object or array of handle objects.

Definition at line 112 of file class_substitutes.c.

11.6.2.12 matlabtypesubstitute handle::permute

Rearranges the dimensions of the handle object array. See the MATLAB permute function.

Definition at line 177 of file class_substitutes.c.

11.6.2.13 matlabtypesubstitute handle::reshape

hanges the dimensions of the handle object array to the specified dimensions. See the MATLAB reshape function.

Definition at line 182 of file class_substitutes.c.

11.6.2.14 matlabtypesubstitute handle::sort

ort the handle objects in any array in ascending or descending order.

The order of handles is purely arbitrary, but reproducible in a given MATLAB session. See the MATLAB sort function.

Definition at line 189 of file class_substitutes.c.

11.6.2.15 matlabtypesubstitute handle::transpose

Transposes the elements of the handle object array.

Definition at line 172 of file class_substitutes.c.

The documentation for this class was generated from the following file:

- /home/ayonga/Dropbox/research/dzopt/direct_hzd_optimization/docs/config/class_substitutes.c

11.7 integer Class Reference

An integer value.

11.7.1 Detailed Description

An integer value.

This class is an artificially created class in doxygen to allow more precise type declarations. Matlab types associated with this class are all int-types (int8, uint8 etc).

The documentation for this class was generated from the following file:

- /home/ayonga/Dropbox/research/dzopt/direct_hzd_optimization/docs/config/class_substitutes.c

11.8 logical Class Reference

A boolean value.

11.8.1 Detailed Description

A boolean value.

This class can be seen as synonym for boolean values/flags used inside classes. In order to stick with matlab conventions/datatypes, this class was named logical instead of bool or boolean.

This class is an artificially created class in doxygen to allow more precise type declarations

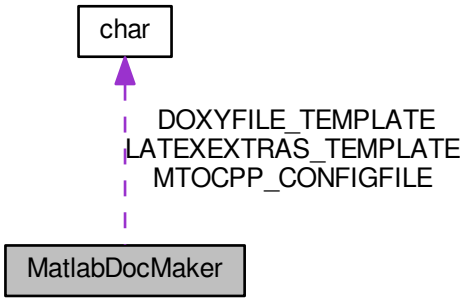
The documentation for this class was generated from the following file:

- /home/ayonga/Dropbox/research/dzopt/direct_hzd_optimization/docs/config/class_substitutes.c

11.9 MatlabDocMaker Class Reference

MatlabDocMaker: Automated documentation creation using doxygen and mtoc++ from within MatLab.

Collaboration diagram for MatlabDocMaker:



Static Public Member Functions

- static mlhsInnerSubst<::char, name > getProjectName ()

  Returns the project name.
- static mlhsInnerSubst<::char, dir > getOutputDirectory ()

  Returns the directory where the applications source files reside.
- static mlhsInnerSubst<::char, dir > getSourceDirectory ()

  Returns the directory where the applications source files reside.
- static mlhsInnerSubst<::char, dir > getConfigDirectory ()

  Returns the directory where the applications documentation configuration files reside.
- static mlhsInnerSubst<::char, desc > getProjectDescription ()

  Returns the short project description.
- static noret::substitute setProjectDescription (::char value)

  Sets the project description.
- static mlhsInnerSubst<::char, version > getProjectVersion ()

  Returns the current version of the project.
- static noret::substitute setProjectVersion (::char value)

  Sets the project version.
- static mlhsInnerSubst< matlabtypesubstitute, fullPath > getProjectLogo ()

  Returns the logo image file for the project. Either an absolute path or a plain filename. For the latter case the image file is assumed to reside inside the directory returned by MatlabDocMaker.getConfigDirectory.
- static noret::substitute setProjectLogo (::char value)

  Sets the project logo. Set to ' to unset.
- static noret::substitute open ()

  Opens the generated documentation.
- static noret::substitute create (matlabtypesubstitute varargin)

  Creates the Doxygen documentation.
- static noret::substitute setup ()

  Runs the setup script for MatlabDocMaker and collects all necessary paths in order for the documentation creation to work properly.

Static Public Attributes

- static const ::char DOXYFILE_TEMPLATE = "Doxyfile.template"

  File name for the doxygen configuration file processed by the MatlabDocMaker.
- static const ::char LATEXEXTRAS_TEMPLATE = "latexextras.template"

  File name for the latex extras style file processed by the MatlabDocMaker.
- static const ::char MTOCPP_CONFIGFILE = "mtocpp.conf"

  File name the mtoc++ configuration file.

11.9.1 Detailed Description

MatlabDocMaker: Automated documentation creation using doxygen and mtoc++ from within MatLab.

Currently documentation creation for unix and windows environment is supported.

Prerequisites

The following tools must be installed and present on the PATH or reside inside the folder returned by MatlabDocMaker.getConfigDirectory.

- `mtocpp`, `mtocpp_post` (the main tool)
- `doxygen` (mtoc++ is a filter for doxygen)

Strongly recommended

- `latex` Doxygen supports built-in latex formulas and MatlabDocMaker/mtoc++ allows for easy extra latex inclusions and notation in code
- `dot` Doxygen creates really nice inheritance graphs and collaboration diagrams with dot.

Author

Daniel Wirtz

Date

2011-10-13

Change in 1.5 (dw, 2013-12-03) Fixed default value selection for properties, now not having set a description or logo does not cause an error to be thrown.

Change in 1.5 (dw, 2013-02-21) Fixed the callback for suggested direct documentation creation after MatlabDocMaker.setup (Thanks to Aurelien Queffurust)

Change in 1.5 (dw, 2013-02-12) Also added the escaping for the Logo file. Thanks to Chris Olien for the hint.

Change in 1.5 (dw, 2013-01-07) Included some backslash escaping for paths on windows platforms. Thanks to MathWorks Pilot Engineer '`Arvind Jayaraman`' for providing the feedback and code!

Change in 1.4 (dw, 2012-10-18) Removed `m4` dependency and included constant properties for configuration file names.

New in 1.4 (dw, 2012-10-16)

- Added two more configuration variables "ProjectDescription" and "ProjectLogo" for easier configuration of the MatlabDocMaker in many cases. Thanks to Wolfgang Mennerich http://www.mathworks.com/matlabcentral/fileexchange/authors/272859 for the suggestion.
- Restructured the configuration, now only the project name function has to be implemented (the preferences tag depends on it, there might be more than one project within the same Matlab installation whos documentation is created using this tool). The rest can be provided either at setup time or later via suitable setters for the version, description and logo.
- Automatically setting HaveDot in the doxygen config whenever its found on the environment path.
- Added basic support for LaTeX documentation creation. Using the parameter `latex`=true for the create method creates the LaTeX version of the documentation in a folder "latex" in the OutputDirectory (default behaviour)

Change in 1.4 (dw, 2012-09-27) Added automatic dot Graphviz tool detection on call to create.

Change in 1.3 (dw, 2012-02-16)

- Now also collecting error messages from mtocpp_post and adding them to the warnings.log file.
- Added the directive "LD_LIBRARY_PATH= " for unix systems, as MatLab sets it inside its executing environment. This can lead to errors if doxygen and/or mtoc++ have been built using never GLIBC (libstd) versions than the one shipped with MatLab.

Change in 1.3 (dw, 2012-01-16)

- Properly using the correct file separators everywhere now
- Hyperlinked the log file so it can be opened directly

Change in 1.3 (dw, 2012-01-14) Not displaying the "generated warnings"-text if there have not been any during documentation creation.

Change in 1.2 (dw, 2011-11-27)

- Included documentation creation for the Windows platform and combined the old methods into one (small effective differences)
- No longer storing the doxygen binary file in the prefs as a lot of tools must be present on the path anyways. The new paradigm is to expect all required 3rd-party programmes to be available on PATH. As backup the configuration files directory is added to the Matlab PATH environment **nonpermanently** and any executables found there will thus also be usable.
- Included checks for `dot` and `latex` at the setup stage to recommend installation of those tools if not present (the default doxygen settings in Doxyfile.m4 are to use both)

Change in 1.2 (dw, 2011-11-08) Improved the createUnix method by displaying the warnings and writing the output to a log file afterwards. Not using cprintf anymore as this is 3rd party software.

Change in 1.2 (dw, 2011-11-07) Fixed a recursion bug caused by copy and paste. Now the preferences are stored on an per-application basis.

Change in 1.2 (dw, 2011-11-04) Changed the name to MatlabDocMaker in order to export it into the mtoc++ distribution later.

New in 1.2 (dw, 2011-10-13) Added this class and moved documentation related stuff here from the KerMor class.

This class is part of the mtoc++ tool

- Homepage http://www.morepas.org/software/mtocpp/
- License http://www.morepas.org/software/mtocpp/docs/licensing.html

Definition at line 17 of file MatlabDocMaker.m.

11.9.2 Member Function Documentation

11.9.2.1 noret::substitute MatlabDocMaker::create (matlabtypesubstitute *varargin*) [static]

Creates the Doxygen documentation.

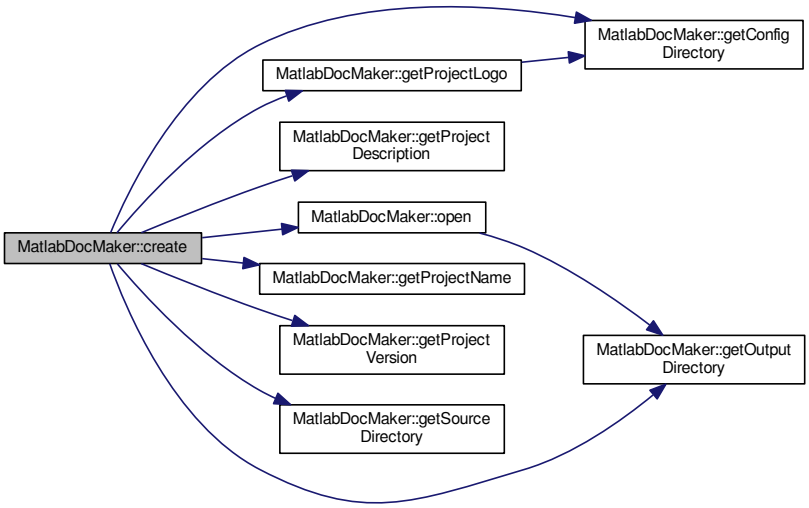
Parameters

varargin	Optional parameters for creation. `create ( [ "open", open_value ] [, "latex", latex_value ] )` *Named Parameters for varargin:* • open Set to true if the documentation should be opened after successful compilation **Default:** false • latex Set to true if LaTeX output should be generated, too. **Default:** false

Definition at line 412 of file MatlabDocMaker.m.

References DOXYFILE_TEMPLATE, getConfigDirectory(), getOutputDirectory(), getProjectDescription(), getProjectLogo(), getProjectName(), getProjectVersion(), getSourceDirectory(), LATEXEXTRAS_TEMPLATE, MTOCPP_CONFIGFILE, and open().

Here is the call graph for this function:



11.9.2.2 mlhsInnerSubst<::char, dir > MatlabDocMaker::getConfigDirectory ( ) [static]

Returns the directory where the applications documentation configuration files reside.

This folder must contain at least the files "mtoc.conf" and "Doxyfile.template"

Return values

dir	The documentation configuration directory

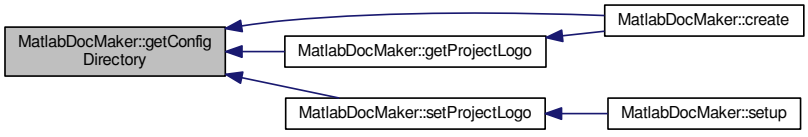
Note

This method has the MATLAB method attribute `Sealed` set to true. It cannot be overwritten.
`matlab documentation of method attributes.`

Definition at line 225 of file MatlabDocMaker.m.

Referenced by create(), getProjectLogo(), and setProjectLogo().

Here is the caller graph for this function:



11.9.2.3 mlhsInnerSubst<::char, dir > MatlabDocMaker::getOutputDirectory ( ) [static]

Returns the directory where the applications source files reside.

Return values

dir	The output directory

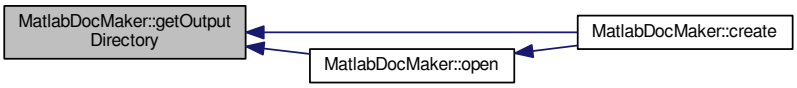
Note

This method has the MATLAB method attribute `Sealed` set to true. It cannot be overwritten.
`matlab documentation of method attributes.`

Definition at line 195 of file MatlabDocMaker.m.

Referenced by create(), and open().

Here is the caller graph for this function:



11.9.2.4 mlhsInnerSubst<::char, desc > MatlabDocMaker::getProjectDescription ( ) [static]

Returns the short project description.

See also

setProjectDescription

Return values

desc	The short project description

Default:

[]

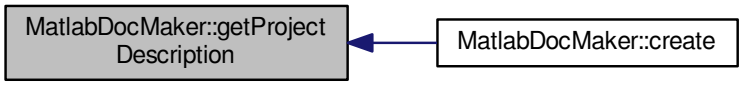
Note

This method has the MATLAB method attribute `Sealed` set to true. It cannot be overwritten.
`matlab documentation of method attributes.`

Definition at line 244 of file MatlabDocMaker.m.

Referenced by create().

Here is the caller graph for this function:



11.9.2.5 mlhsInnerSubst< matlabtypesubstitute, fullPath > MatlabDocMaker::getProjectLogo ( ) [static]

Returns the logo image file for the project. Either an absolute path or a plain filename. For the latter case the image file is assumed to reside inside the directory returned by MatlabDocMaker.getConfigDirectory.

See also

setProjectLogo

Return values

logoFile	The projects logo image file.

Default:

[]

Note

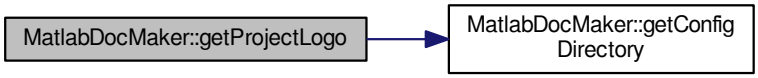
This method has the MATLAB method attribute `Sealed` set to true. It cannot be overwritten.
`matlab documentation of method attributes.`

Definition at line 317 of file MatlabDocMaker.m.

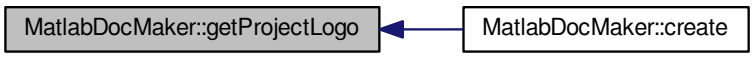
References getConfigDirectory().

Referenced by create().

Here is the call graph for this function:



Here is the caller graph for this function:



11.9.2.6 mlhsInnerSubst<::char, name > MatlabDocMaker::getProjectName ( ) [static]

Returns the project name.

Note

Changing the return value of this method will require another execution of MatlabDocMaker.setup as the preferences storage key also depends on it.

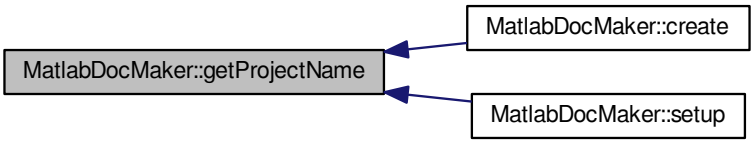
Return values

name	The project name

Definition at line 170 of file MatlabDocMaker.m.

Referenced by create(), and setup().

Here is the caller graph for this function:



11.9.2.7 mlhsInnerSubst<::char, version > MatlabDocMaker::getProjectVersion ( ) [static]

Returns the current version of the project.

Note

The built-in @new and @change tags from the Doxyfile.template support two-level versioning a la X.X.

See also

setProjectVersion

Return values

version	The project version

Default:

[]

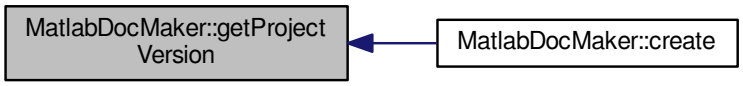
Note

This method has the MATLAB method attribute `Sealed` set to true. It cannot be overwritten.
`matlab documentation of method attributes.`

Definition at line 279 of file MatlabDocMaker.m.

Referenced by create().

Here is the caller graph for this function:



11.9.2.8 mlhsInnerSubst<::char, dir > MatlabDocMaker::getSourceDirectory ( ) [static]

Returns the directory where the applications source files reside.

Return values

dir	The project source directory

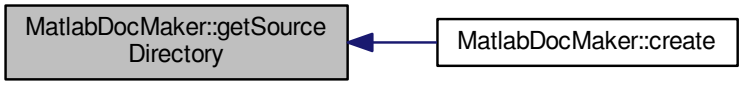
Note

This method has the MATLAB method attribute `Sealed` set to true. It cannot be overwritten.
`matlab documentation of method attributes.`

Definition at line 210 of file MatlabDocMaker.m.

Referenced by create().

Here is the caller graph for this function:



11.9.2.9 noret::substitute MatlabDocMaker::open ( ) [static]

Opens the generated documentation.

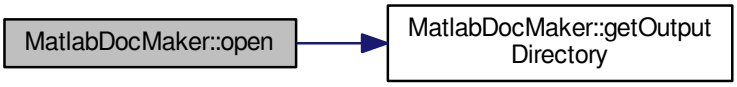
Depending on the systems type the generated index.html is opened in the systems default browser.

Definition at line 391 of file MatlabDocMaker.m.

References getOutputDirectory().

Referenced by create().

Here is the call graph for this function:



Here is the caller graph for this function:

11.9.2.10 noret::substitute MatlabDocMaker::setProjectDescription (::char *value*) [static]

Sets the project description.

See also

getProjectDescription

Parameters

value	The description

Note

This method has the MATLAB method attribute `Sealed` set to true. It cannot be overwritten.
`matlab documentation of method attributes.`

Definition at line 260 of file MatlabDocMaker.m.

11.9.2.11 noret::substitute MatlabDocMaker::setProjectLogo (::char *value*) [static]

Sets the project logo. Set to ' to unset.

See the doxygen documentation for valid logo file types (wont be checked here).

See also

getProjectLogo

Parameters

value	The logo file to use. Must be either an absolute path or a plain filename, in which case the image is assumed to reside inside the MatlabDocMaker.getConfigDirectory directory.

Note

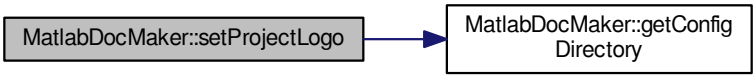
This method has the MATLAB method attribute `Sealed` set to true. It cannot be overwritten.
`matlab documentation of method attributes.`

Definition at line 347 of file MatlabDocMaker.m.

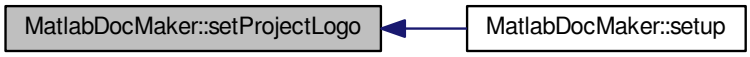
References getConfigDirectory().

Referenced by setup().

Here is the call graph for this function:



Here is the caller graph for this function:



11.9.2.12 noret::substitute MatlabDocMaker::setProjectVersion (::char *value*) [static]

Sets the project version.

See also

getProjectVersion

Parameters

value	The version string

Note

This method has the MATLAB method attribute `Sealed` set to true. It cannot be overwritten.
`matlab documentation of method attributes.`

Definition at line 298 of file MatlabDocMaker.m.

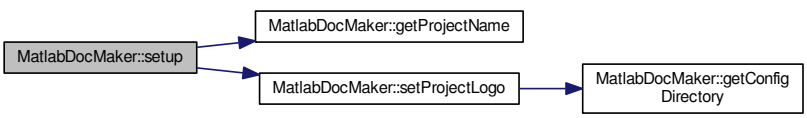
11.9.2.13 noret::substitute MatlabDocMaker::setup ( ) [static]

Runs the setup script for MatlabDocMaker and collects all necessary paths in order for the documentation creation to work properly.

Definition at line 631 of file MatlabDocMaker.m.

References getProjectName(), and setProjectLogo().

Here is the call graph for this function:



11.9.3 Member Data Documentation

11.9.3.1 MatlabDocMaker::DOXYFILE_TEMPLATE = "Doxyfile.template" [static]

File name for the doxygen configuration file processed by the MatlabDocMaker.

Assumed to reside in the MatlabDocMaker.getConfigDirectory

Default: `Doxyfile.template`

Definition at line 130 of file MatlabDocMaker.m.

Referenced by create().

11.9.3.2 MatlabDocMaker::LATEXEXTRAS_TEMPLATE = "latexextras.template" [static]

File name for the latex extras style file processed by the MatlabDocMaker.

Assumed to reside in the MatlabDocMaker.getConfigDirectory. If not found, no latex extras are used.

Default: `latexextras.template`

Definition at line 142 of file MatlabDocMaker.m.

Referenced by create().

11.9.3.3 MatlabDocMaker::MTOCPP_CONFIGFILE = "mtocpp.conf" [static]

File name the mtoc++ configuration file.

Assumed to reside in the MatlabDocMaker.getConfigDirectory. If not found, no special configuration is used.

Default: `mtocpp.conf`

Definition at line 155 of file MatlabDocMaker.m.

Referenced by create().

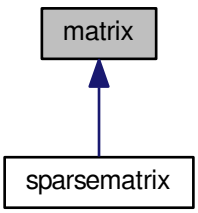
The documentation for this class was generated from the following file:

- MatlabDocMaker.m

11.10 matrix Class Reference

A matlab matrix.

Inheritance diagram for matrix:



11.10.1 Detailed Description

A matlab matrix.

This class is an artificially created class in doxygen to allow more precise type declarations

Definition at line 73 of file class_substitutes.c.

The documentation for this class was generated from the following file:

- /home/ayonga/Dropbox/research/dzopt/direct_hzd_optimization/docs/config/class_substitutes.c

11.11 rowvec Class Reference

A matlab row vector.

11.11.1 Detailed Description

A matlab row vector.

This class is an artificially created class in doxygen to allow more precise type declarations

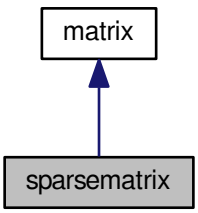
The documentation for this class was generated from the following file:

- /home/ayonga/Dropbox/research/dzopt/direct_hzd_optimization/docs/config/class_substitutes.c

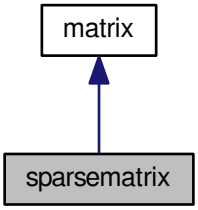
11.12 sparsematrix Class Reference

A matlab sparse matrix.

Inheritance diagram for sparsematrix:



Collaboration diagram for sparsematrix:



11.12.1 Detailed Description

A matlab sparse matrix.

This class is an artificially created class in doxygen to allow more precise type declarations

Definition at line 81 of file class_substitutes.c.

The documentation for this class was generated from the following file:

- /home/ayonga/Dropbox/research/dzopt/direct_hzd_optimization/docs/config/class_substitutes.c

11.13 struct Class Reference

A MatLab struct.

11.13.1 Detailed Description

A MatLab struct.

This class is an artificially created class in doxygen to allow more precise type declarations

The documentation for this class was generated from the following file:

- /home/ayonga/Dropbox/research/dzopt/direct_hzd_optimization/docs/config/class_substitutes.c

11.14 varargin Class Reference

A variable number of input arguments.

11.14.1 Detailed Description

A variable number of input arguments.

This class is an artificially created class in doxygen to allow more precise type declarations.

For more information about the varargin argument see the MatLab documentation on varargin.

The documentation for this class was generated from the following file:

- /home/ayonga/Dropbox/research/dzopt/direct_hzd_optimization/docs/config/class_substitutes.c

11.15 varargout Class Reference

A variable number of output arguments.

11.15.1 Detailed Description

A variable number of output arguments.

This class is an artificially created class in doxygen to allow more precise type declarations.

For more information about the varargout argument see the MatLab documentation on varargout.

The documentation for this class was generated from the following file:

- /home/ayonga/Dropbox/research/dzopt/direct_hzd_optimization/docs/config/class_substitutes.c

12 File Documentation

12.1 /home/ayonga/Dropbox/research/dzopt/direct_hzd_optimization/docs/config/class_substitutes.c File Reference

Classes

- class matrix
  A matlab matrix.
- class sparsematrix
  A matlab sparse matrix.
- class handle
  Matlab's base handle class (documentation generation substitute)

12.2 /home/ayonga/Dropbox/research/dzopt/direct_hzd_optimization/docs/config/developers.c File Reference

12.3 MatlabDocMaker.m File Reference

Classes

- class MatlabDocMaker
  MatlabDocMaker: Automated documentation creation using doxygen and mtoc++ from within MatLab.

Index