



phpdocgen

Section: phpdocgen (1)

NAME

phpdocgen - A perl script that generates an API documentation for PHP source files

SYNOPSIS

phpdocgen_checker [options] *file* [*file* ...]

DESCRIPTION

phpdocgen_checker is a script which extract the data in the same way as phpdocgen, and display the result of the coherence checking algorithm.

OPTIONS

--defpack *name*

Sets the name of the default package.

--doctitle *text*

Sets the title of the documentation that appears in the summary.

--exclude *file*

Excludes the *file* from the list of files which will be parsed.

--exclude-from *file*

Excludes from the list of files which will be parsed all the files which are listed inside the specified *file*. The *file* must contain one filename per line.

--extscanner *name*

Force phpdocgen to use the specified extended scanner. This option is not officially supported. The *name* must be the name of the Perl class which must be used to scan the PHP source files.

-f

--force

If the option **-o** was not given, forces to overwrite into the default output directory (./doc/).

--filters *filters*

Sets the filters used to recognized the PHP files. *filters* must be a list of filters separated by ':'. You can use the shell wildcards in the filters. The default filter is '*.php*'.
-?

-h

--help

Show the list of available options.

--html

Generates the documentation as HTML pages.

--lang *name*

Sets the language used by the documentation generator.

--langlist

Display the list of supported languages.

--latex

Generates the documentation as a LaTeX document.

--man

--manual

Show the manual page.

-o *file*

--output *file*

Sets the directory or the file in which the documentation will be put.

--[no]phpgen

Indicates if the documentation of the PHP functions and classes must be generated.

-q

Don't be verbose: only error messages are displayed.

-r

-R

Scans the directories recursively.

--[no]sources

Generates additional pages that contain the PHP source code.

--theme *name*

Sets the graphical theme used by the documentation generator.

--themelist

Display the list of supported themes.

-v

Be more verbose.

--verbatim

All comments have the tag `@verbatim` by default.

--version

Show the version of this script.

--[no]warning

If false, the warning are converted to errors.

--[no]webgen

Indicates if the documentation of the web modules and pages.

--windowtitle *text*

Sets the title of the documentation that appears as the window's title.

-x *file*

See **--exclude**

-X *file*

See **--exclude-from** .

Supported tags

@attribute [public|protected|private] *type name*

This optional tag is to force the current comment to be for for an class attribute named *name*. You could use the tag `@class` to specified the class in which this attribute is. *type* must be one of the types described in the section **Types**. If this tag is missed, the type is always ```mixed```. The modifiers ```public```, ```protected``` and ```private``` are not supported by PHP. They are added for documentation convenience.

@author *name*

Sets the *name*name of an author.

{@block *text*}

The *text* is considered as an unsplitable text.

@brief *comment*

Sets the brief description for the comment. If this tag was not given, the first sentence of the description will be the brief description.

@bug [fixed] *comment*

Adds a bug report to the current comment. It is explained by the *comment* . If ```fixed``` is present, it applies that this bug was fixed.

@class *name*

This optional tag is used to force the classname used by a comment.

@constant [*type*] *name*

This optional tag is used to force the comment to be for a global constant named *name*. You could use the tag `@package` to specified the package in which this constant is. *type* must be one of the types described in the section **Types**.

@constructor *name*

This optional tag is for a constructor of the class named *name*. You could use the tag `@class` to specified the class in which this attribute is.

@copyright *comment*

Adds a copyright description.

@date *date*

Attaches a date to the current comment.

@deprecated *text*

The currently documented stuff is deprecated. *text* is the explanation.

{@example [*frame|noframe*] *file|text*}

Adds a example inside the text. It could be enclosed inside a frame box (option *frame*) or not (option *noframe*). The example could be typed as inline *text* or put inside a *file*.

@extends *name*

See `@inherited`

@function *name*

This optional tag is used to force the current comment to be for a global function named *name*. You could use the tag `@package` to specified the class in which is this attribute.

@global *name*

See `@use`.

{@hash [*key*] *comment*}

Not supported yet.

@inherited *name*

This optional tag permits to force the documented class inherits from the class *name*.

{@link *name* [*comment*]}

Adds a link to the documentation of *name* with the *comment*.

@log *date comment*

Adds a changelog for the current comment. The change is explained by the *comment* and was applied since the *date*.

@method [*static*] [*public|protected|private*] *name*

This optional tag is used to force the current comment to be for an class method named *name*. You could use the tag `@class` to specified the class in which this method is. If ```static``` is present, it applies that this method is static, i.e. the use of the variable `$this` is not allowed. The modifiers ```public```, ```protected``` and ```private``` are not supported by PHP. They are added for documentation convenience.

@modifiers [*static*] [*private|protected|public*]

This optional tag is used to specify the modified of the current commented object. It override the modifiers given by `@method` and `@attribute`.

@package *name*

Forces the current package name to *name*.

@param [*optional*] [*reference|ref*] *type name comment*

Adds a parameter comment for the current documented stuff. The parameter has named *name* and is commented by *comment*. *type* must be one of the types described in the section **Types**. If ```optional``` was present, it applies that this parameter is optional. If ```reference``` (or ```ref```) was present, it applies that this parameter is passed by reference.

@private

Indicates that the current method or attribute has a private access.

@protected

Indicates that the current method or attribute has a protected access.

@public

Indicates that the current method or attribute has a public access.

@return [reference|ref] [*type*] *comment*

Explains what is return by the current documented function. *type* must be one of the types described in the section **Types**. If ``reference" (or ``ref") was present, it applies that the returned value is a reference.

@see *comment*

Gives a pointer to another documentation. If *comment* is a valid name, show a link to the corresponding documentation page.

@since *comment*

Explains from when the current comment is exists.

@static

Indicates that the current method or attribute was static.

@todo *comment*

Explains something to do for the current documented object.

@use *name*

Indicates that the current function uses the global variable *name*.

@variable *type name*

This optional tag permits to force the current comment to be for a global variable named *name*. You could use the tag `@package` to specified the package in which this variable is. *type* must be one of the types described in the section **Types**.

@verbatim

The explanation of the current comment will be putted as-is in the generated documentation. The generated explanation will be enclosed by `<PRE>` and `</PRE>`.

@version *comment*

Describes the version of the documented object.

@webmodule *name*

This tag permits to force the current comment to be for a webmodule called *name*. The *name* must be a valid HTML path.

@webpage [*path*] / *name*

This tag permits to force the current comment to be for a webpage called *name*. The *path* must be a valid HTML path.

Types

array

is an array in which all the keys are integers.

bool

boolean

supports only the values **TRUE** and **FALSE**.

callback

is an user callback function (eg, parameter of `usort()`).

flt

float

is for floating-point numbers.

hash

hashtable

is for hashtables.

int

integer

is for integer numbers.

mix

mixed

indicates that the value could be anything (one of the other types).

num

number

is a number ie, a float or an integer.

obj

object

is for PHP objects.

resource

is an external resource (eg, SQL request).

str

string

is a string.

time

date

timestamp

an unix timestamp.

COPYRIGHT

Copyright (c) 2002-03 Stéphane Galland <galland@arakhne.org>, under GPL.

SEE ALSO

javadoc

Generated by kio_man, KDE version 4.4.5 (KDE 4.4.5)