

# Deployment Test

0.7.0

Generated by Doxygen 1.9.1

<b>1 Main Page</b>	<b>1</b>
<b>1 Main Page</b>	<b>1</b>
1.1 Deployment test	1
1.1.1 Setup	2
1.1.2 TODO	2
<b>2 Change Log</b>	<b>2</b>
2.1 0.7.0	2
2.2 Change Log	2
2.2.1 0.6.0	2
2.2.2 0.1.0	2
<b>3 Namespace Index</b>	<b>2</b>
3.1 Namespace List	2
<b>4 Class Index</b>	<b>2</b>
4.1 Class List	2
<b>5 File Index</b>	<b>3</b>
5.1 File List	3
<b>6 Namespace Documentation</b>	<b>3</b>
6.1 deploytest Namespace Reference	3
6.1.1 Detailed Description	3
<b>7 Class Documentation</b>	<b>3</b>
7.1 deploytest::TestClass Class Reference	3
7.1.1 Detailed Description	4
7.1.2 Constructor & Destructor Documentation	4
7.1.3 Member Function Documentation	4
<b>8 File Documentation</b>	<b>5</b>
8.1 include/test.hpp File Reference	5
8.1.1 Detailed Description	5
8.1.2 Macro Definition Documentation	6
<b>Index</b>	<b>7</b>

# 1 Main Page

## 1.1 Deployment test

A test repository to test GitHub Actions to do the following:

- Create release and publish Doxygen HTML documentation to a version specific subdirectory in the `gh-pages` branch
- Publish the Doxygen for the *latest* (development) code to the `latest` subdirectory in the `gh-pages` branch
- Create branches for patches to major.minor release builds

### 1.1.1 Setup

Create the `gh-pages` branch for a repository:

```
git checkout --orphan gh-pages
git reset --hard
touch .nojekyll
git push origin gh-pages
```

### 1.1.2 TODO

Especially in the `create-release` action with have race conditions and and possible problems on push

## 2 Change Log

### 2.1 0.7.0

Added `--ignore-unmatch` to `git rm`

### 2.2 Change Log

#### 2.2.1 0.6.0

Fixed ( to (

#### 2.2.2 0.1.0

Fixed the fake issue Fixed the rc2 build issue

**2.2.2.1 Known Issues** Not fully tested

## 3 Namespace Index

### 3.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

**deploytest**

The Deployment Test namespace

**3**

## 4 Class Index

### 4.1 Class List

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

[deploytest::TestClass](#)  
The Deployment Test struct

3

## 5 File Index

### 5.1 File List

Here is a list of all documented files with brief descriptions:

[include/test.hpp](#)  
The main Deployment Test file

5

## 6 Namespace Documentation

### 6.1 deploytest Namespace Reference

The Deployment Test namespace.

#### Classes

- class [TestClass](#)  
*The Deployment Test struct.*

#### 6.1.1 Detailed Description

The Deployment Test namespace.

## 7 Class Documentation

### 7.1 deploytest::TestClass Class Reference

The Deployment Test struct.

```
#include <test.hpp>
```

#### Public Member Functions

- [TestClass](#) ()  
*Creates an empty Test Class object.*
- [TestClass](#) (const **std::string** &name)  
*Creates a Test Class with the specified name.*
- const **std::string** & [Name](#) () const  
*Gets the name.*
- **std::size\_t** [Length](#) () const  
*Gets the object length.*

### 7.1.1 Detailed Description

The Deployment Test struct.

### 7.1.2 Constructor & Destructor Documentation

#### 7.1.2.1 **TestClass()** [1/2] `deploytest::TestClass::TestClass ( )`

Creates an empty Test Class object.

#### 7.1.2.2 **TestClass()** [2/2] `deploytest::TestClass::TestClass ( const std::string & name )`

Creates a Test Class with the specified name.

##### Parameters

<i>name</i>	The name for the class
-------------	------------------------

### 7.1.3 Member Function Documentation

#### 7.1.3.1 **Length()** `std::size_t deploytest::TestClass::Length ( ) const [inline]`

Gets the object length.

##### Returns

The length of the object

#### 7.1.3.2 **Name()** `const std::string& deploytest::TestClass::Name ( ) const [inline]`

Gets the name.

##### Returns

The name of the object

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

- [include/test.hpp](#)

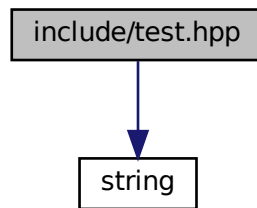
## 8 File Documentation

### 8.1 include/test.hpp File Reference

The main Deployment Test file.

```
#include <string>
```

Include dependency graph for test.hpp:



#### Classes

- class [deploytest::TestClass](#)  
*The Deployment Test struct.*

#### Namespaces

- [deploytest](#)  
*The Deployment Test namespace.*

#### Macros

- `#define` [DEPLOYTEST\\_VERSION\\_MAJOR](#) 0  
*Major version number of the Library.*
- `#define` [DEPLOYTEST\\_VERSION\\_MINOR](#) 7  
*Minor version number of the Library.*
- `#define` [DEPLOYTEST\\_VERSION\\_REVISION](#) 0  
*Revision version number of the Library.*
- `#define` [DEPLOYTEST\\_VERSION\\_SUFFIX](#) ""  
*Pre-release suffix of the Library.*

#### 8.1.1 Detailed Description

The main Deployment Test file.

## 8.1.2 Macro Definition Documentation

### 8.1.2.1 DEPLOYTEST\_VERSION\_MAJOR `#define DEPLOYTEST_VERSION_MAJOR 0`

Major version number of the Library.

### 8.1.2.2 DEPLOYTEST\_VERSION\_MINOR `#define DEPLOYTEST_VERSION_MINOR 7`

Minor version number of the Library.

### 8.1.2.3 DEPLOYTEST\_VERSION\_REVISION `#define DEPLOYTEST_VERSION_REVISION 0`

Revision version number of the Library.

### 8.1.2.4 DEPLOYTEST\_VERSION\_SUFFIX `#define DEPLOYTEST_VERSION_SUFFIX ""`

Pre-release suffix of the Library.

Use `semantic versioning` suffices with alpha characters are compared by ASCII ordering; therefore `-DEV` `< -alpha < -beta < -rc` providing `DEV` is uppercase and `alpha`, `beta`, and `rc` are all lower case.

## Index

- deploytest, [3](#)
- deploytest::TestClass, [3](#)
  - Length, [4](#)
  - Name, [4](#)
  - TestClass, [4](#)
- DEPLOYTEST\_VERSION\_MAJOR
  - test.hpp, [6](#)
- DEPLOYTEST\_VERSION\_MINOR
  - test.hpp, [6](#)
- DEPLOYTEST\_VERSION\_REVISION
  - test.hpp, [6](#)
- DEPLOYTEST\_VERSION\_SUFFIX
  - test.hpp, [6](#)
- include/test.hpp, [5](#)
- Length
  - deploytest::TestClass, [4](#)
- Name
  - deploytest::TestClass, [4](#)
- test.hpp
  - DEPLOYTEST\_VERSION\_MAJOR, [6](#)
  - DEPLOYTEST\_VERSION\_MINOR, [6](#)
  - DEPLOYTEST\_VERSION\_REVISION, [6](#)
  - DEPLOYTEST\_VERSION\_SUFFIX, [6](#)
- TestClass
  - deploytest::TestClass, [4](#)