

# Doxygen



Гвоздарёв Сергей  
MDSP Project Intel Lab  
MIPT

*Апрель 2011*

# Что такое Doxygen?

- Doxygen – это кроссплатформенная система автоматического документирования кода с поддержкой языков C++, C, Java, Objective-C, PHP, C# etc
- Doxygen умеет анализировать исходный код проекта и создавать удобную документацию в формате HTML, Latex, RTF, XML, man, CHM.

# Комментирование в стиле Doxygen

- /\*\*

Комментарии

\*/

- ///  
Комментарий

# Директивы

- **@brief** - краткое описание
- **@details** - подробное описание, то что будет показано, когда вы нажмете "Подробнее".

# Директивы

- **@param** - параметры передаваемые функции.
- **@return** - возвращаемое функцией значение
- **@throw** - исключения выбрасываемые функцией.

# Директивы

- **@file** - имя файла
- **@author** - автор
- **@version** - версия
- **@section** – начало раздела, например  
LICENSE или DESCRIPTION

# Директивы на уровне проекта

- **@mainpage** - основная страница проекта, то с чего начинается просмотр документации.
- **@page** - дополнительные страницы проекта
- **@ref** - ссылки на страницы проекта.

# Примеры

```
/**  
 *@brief Class of simulator's core description  
 *@details Подробное описание....  
 *  
 */  
class Core : public log
```

## MDSP r561

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<b><a href="#">Classes</a></b>	<a href="#">Files</a>
<a href="#">Class List</a>	<a href="#">Class Index</a>	<a href="#">Class Members</a>	
<b>Core Class Reference</b>			

Class of simulator's core description. [More...](#)

```
#include <core.h>
```

List of all members.

# Примеры

```
/**  
 *Initialize the functional simulator.  
 * @param start_pc Sets up start PC address.  
 * @return void  
 */  
void init ( hostUInt16 start_pc );
```

```
void Core::init ( hostUInt16 start_pc )
```

Initialize the functional simulator.

**Parameters:**

**start\_pc** Sets up start PC address.

**Returns:**

void

Definition at line **28** of file **core.cpp**.

# Примеры

```
/**
 * @mainpage MDSP project Intel Lab at MIPT
 * @ref core fields and methods used to run the simulator
 */
/**
 * @page core core
 * .....
 * @file core.h Header of Core class which defines
 * @author Pavel Zaichenkov
 * @version 1.0
 * @section COPYRIGHT
 * 2009 MDSP team
 * @section DESCRIPTION
 * fields and methods used to run the simulator
 */
```

[Main Page](#)

[Related Pages](#)

[Classes](#)

[Files](#)

## MDSP project Intel Lab at MIPT

[core](#) fields and methods used to run the simulator

## core.h File Reference

```
#include <cassert>
#include <fstream>
#include "log.h"
#include "types.h"
```

[Go to the source code of this file.](#)

### Classes

---

class	<b>Core</b>
	Class of simulator's core description.

---

### Detailed Description

Header of **Core** class which defines

**Author:**

Pavel Zaichenkov

**Version:**

1.0

### COPYRIGHT

2009 MDSP team

### DESCRIPTION

fields and methods used to run the simulator

Definition in file **core.h**.