

Doxygen 사용법

학습목표

- ❖ Doxygen이 무엇인지 설명할 수 있다.
- ❖ Doxygen을 왜 써야하는지 설명할 수 있다.
- ❖ Doxygen을 위한 주석 문법을 이해한다.
- ❖ Doxygen 도구 설정 세팅 방법을 따라한다.

Doxygen이 뭔가요?

Doxygen은 소프트웨어 레퍼런스 문서 생성기입니다.

문서를 왜 만들어야 하죠?

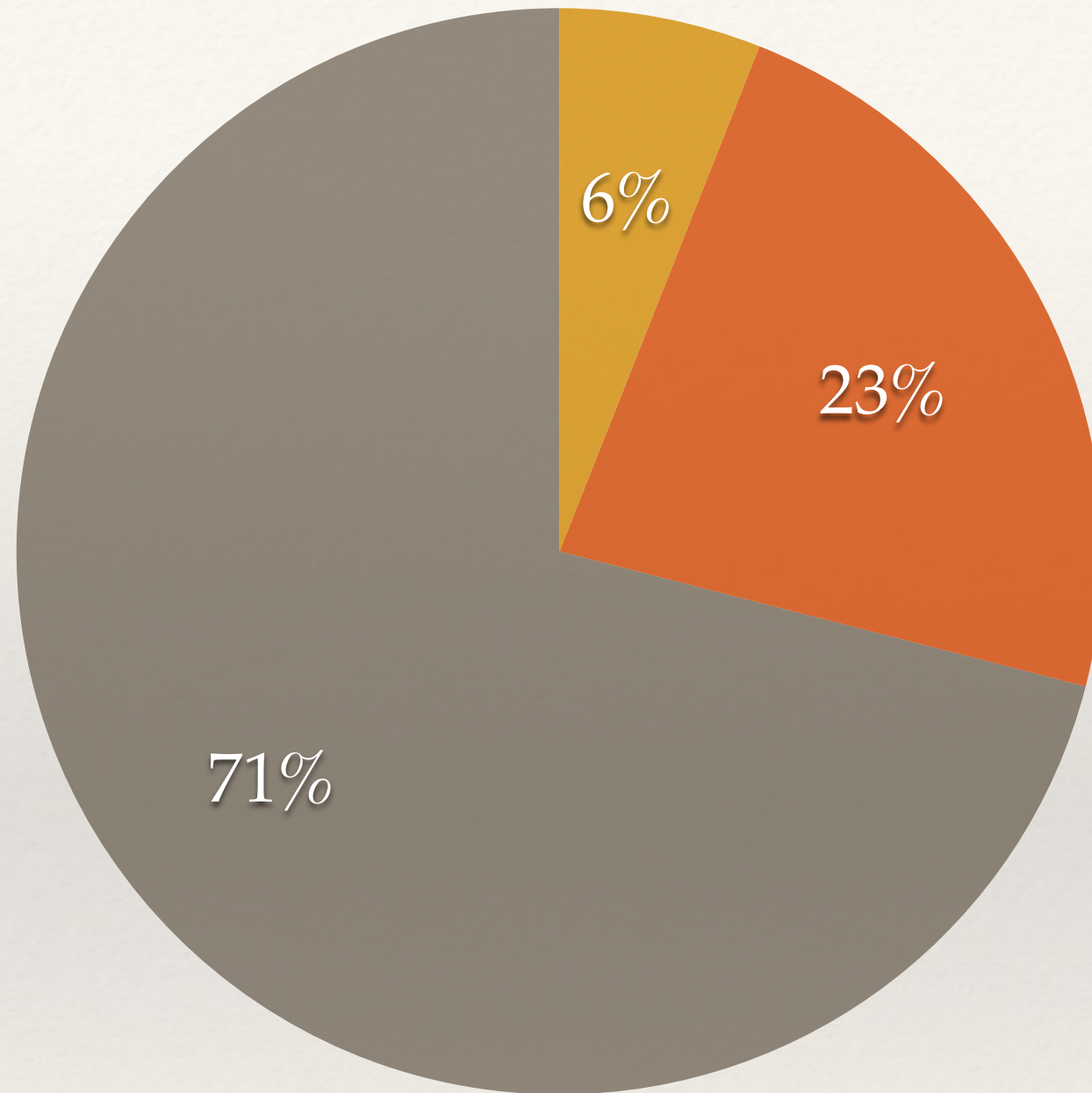
인간 기억력의 한계 때문입니다.

프로그래머는 대부분의 시간을 코드를 이해하는데 소비합니다.

● 코드 생성

● 코드 수정

● 코드 이해



개발 완료 후 몇 주가 지나 버그가 발견됐습니다.
코드를 고치려니 내가 어떻게 구현했는지 기억이 안나네요.

코드를 읽으니 이전의 기억이 하나 둘씩 되살아 납니다.
이제 버그를 수정하는 건 일도 아니에요.

이런 일은 이후에도 계속 반복됩니다.

시간이 흐르면서 코드의 복잡도는 증가했습니다.
나중에는 몇일을 보아도 구조를 잘 모르는 일이 생기고 말았어요.

어떻게 하면 기억을 좀 더 쉽게 되살릴 수 있을까요?

문서를 작성하면 효과가 있습니다.
문서를 읽으면 코드를 훨씬 빨리 파악할 수 있어요.

이젠,

완성된 코드와 더불어 문서도 소중한 결과물이 입니다.

하지만 문서작성도 비용아닌가요?

맞아요. 문서 작성에도 큰 비용이 소모됩니다.
하지만 코드가 문서 자체가 된다면 어떨까요?

코드 버전별로 문서 버전을 관리할 필요가 없습니다.
코드만 관리하면 문서 관리까지 되는거예요.

어떻게 그런 일이 가능하죠?

```
package server;

import ...

/**
 * 이벤트가 발생가 발생할 때, 이벤트를 디코딩하여 적절한 핸들러에 전달한다.
 */
public class Demultiplexer {
    private ServerSocket serverSocket;

    /**
     * 생성자. 서버 소켓을 생성한다.
     * @param
     * @throws IOException
     * @return Nothing
     */
    public Demultiplexer() throws IOException {
        serverSocket = new ServerSocket(5000);
    }

    /**
     * 이벤트가 발생할 때, 디코딩하여 적절한 핸들러에 전달한다.
     * @param handlers Reactor에 등록된 핸들맵 객체.
     * @throws IOException
     * @return Nothing
     */
    public void select(HandleMap handlers) throws IOException {
        Socket socket = serverSocket.accept();
        InputStream inputStream = socket.getInputStream();

        byte[] buffer = new byte[6];
        inputStream.read(buffer);
        String header = new String(buffer);

        buffer = new byte[1024];
        inputStream.read(buffer);
        String body = new String(buffer);

        handlers.get(header).handleEvents(body);
    }
}
```



Public 멤버 함수

Demultiplexer () throws IOException

void **select** (HandleMap handlers) throws IOException

Private 속성

ServerSocket **serverSocket**

상세한 설명

이벤트가 발생가 발생할 때, 이벤트를 디코딩하여 적절한 핸들러에 전달한다.

Demultiplexer.java 파일의 11 번째 라인에서 정의되었습니다.

생성자 & 소멸자 문서화

server.Demultiplexer.Demultiplexer () throws IOException

생성자. 서버 소켓을 생성한다.

매개변수

@throws IOException

Doxygen 주석 문법을 사용하여
코드에 주석을 달면 됩니다.



어서 그 문법을 알려주세요!

네, 알려 드리겠습니다.
따라오세요.

기본적인 **Doxygen** 주석 형태입니다.

/**

* 기본 주석 형태

*/

메인 페이지 주석 예시

```
/**
```

```
 * @mainpage 메인 페이지 제목
```

```
 * 메인 페이지 설명. 블라블라
```

```
 *
```

```
 */
```

파일 주석 예시

```
/**  
 * @file NewClass.java  
 * @brief   간략한 설명  
 * @details 자세한 설명  
 */
```


클래스 주석 예시

```
/**  
 * @author Dongkuk Kim, dongkuk5411@nhnnnext.org  
 * @date    2013-12-31  
 * @version 0.0.1  
 */  
class ClassName extends ExtendedClassName  
{  
    /* do nothing */  
}
```

메서드 주석 예시

```
/**
 * @brief   메서드 간략 설명.
 * @details 메서드 자세히 설명.
 * @param   args 콘솔 파라미터
 * @return  프로그램 상태
 *
 * @bug     메모리 누수 있음.
 * @todo    버그 해결 해야함.
 * @exception StringIndexOutOfBoundsException
 *
 * @see     NewClass
 * @see     http://java.sun.com
 */
public static int main(String[] args)
{
    ...
    return( status );
}
```

@item들은 뭔가요?

주석에 대한 메타데이터입니다.
다음 페이지에서부터 각 아이템들을 소개해 드릴게요.

@mainpage 프로그램의 전체 개요 등을 설명합니다.

@section 프로그램의 개요 별로 설명합니다.

@version 프로그램 버전을 설명합니다.

@breif 간략한 설명을 쓸 때 사용합니다.
@details 자세한 설명을 쓸 때 사용합니다.

@file 파일 이름을 구별할 때 사용합니다.
@author 작성자 이름을 나타낼 때 사용합니다.
@date 작성날짜를 나타낼 때 사용합니다.

@param 함수 파라미터를 설명합니다.
@return 함수의 리턴값을 설명합니다.

`@exeception` 예외 처리를 설명합니다.
`@throws throw`하는 객체나 변수 등을 설명합니다.

@see 참고할 함수나 페이지를 지정합니다.
@todo 해야할 일에 대해서 따로 리스트를 생성합니다.
@bug 버그에 대해서 따로 리스트를 생성합니다.

@code 중요코드를 설명할때 시작 지점 가리킵니다.
@endcode 중요코드 설명할때 종료 지점 가리킵니다.

이로써 @item에 대한 설명은 그만 하겠습니다. 끝이 안보이네요.
<http://www.stack.nl/~dimitri/doxygen/manual/index.html>
에서 더 찾아보세요.

그럼 이제 Doxygen을 주세요.

잠깐만요!



Doxygen 설치에 앞서,
함수 구조도, 클래스 구조도 등의 그래프를 문서에 포함시키기 위해
Graphviz를 설치하는게 좋습니다.

<http://www.graphviz.org/Download..php> 에서
해당 OS에 맞는 Graphviz를 설치해주세요.

<http://www.graphviz.org/pub/graphviz/stable/macos/mountainlion/graphviz-2.34.0.pkg>
맥용 Graphviz 바로 다운로드

그 다음,

<http://www.stack.nl/~dimitri/doxygen/download.html#latestsrc>

에서 해당 OS에 맞는 Doxygen을 설치해주세요.

<http://ftp.stack.nl/pub/users/dimitri/Doxygen-1.8.6.dmg>

맥용 바로 다운로드



Doxygen

실행

이제 Doxygen 설정 세팅법을 보여드릴 겁니다.
한걸음 한걸음 따라와요.

The screenshot shows the 'Doxygen GUI frontend' window. It has a title bar with standard macOS window controls (red, yellow, green buttons). The main content area is divided into two sections. The top section is titled 'Step 1: Specify the working directory from which doxygen will run' and contains a single text input field and a 'Select...' button. The bottom section is titled 'Step 2: Configure doxygen using the Wizard and/or Expert tab, then switch to the Run tab to generate the documentation'. It features three tabs: 'Wizard' (selected), 'Expert', and 'Run'. On the left side of the 'Wizard' tab is a 'Topics' list with items: 'Project', 'Mode', 'Output', and 'Diagrams'. The main area of the 'Wizard' tab is titled 'Provide some information about the project you are documenting' and contains several form fields: 'Project name:' with the value 'My Project', 'Project synopsis:', 'Project version or id:', and 'Project logo:' with a 'Select...' button. Below these is a section titled 'Specify the directory to scan for source code' with a 'Source code directory:' field and a 'Select...' button, followed by an unchecked checkbox for 'Scan recursively'. The final section is titled 'Specify the directory where doxygen should put the generated documentation' with a 'Destination directory:' field and a 'Select...' button. At the bottom of the window are two buttons: 'Previous' and 'Next'.

Doxygen GUI frontend

Step 1: Specify the working directory from which doxygen will run

Step 2: Configure doxygen using the Wizard and/or Expert tab, then switch to the Run tab to generate the documentation

Wizard Expert Run

Topics

- Project
- Mode
- Output
- Diagrams

Provide some information about the project you are documenting

Project name: My Project

Project synopsis:

Project version or id:

Project logo: Select...

Specify the directory to scan for source code

Source code directory: Select...

☐ Scan recursively

Specify the directory where doxygen should put the generated documentation

Destination directory: Select...

Previous Next

프로젝트 루트 폴더를 지정합니다.

The screenshot shows the 'Doxygen GUI frontend' window. At the top, it says 'Step 1: Specify the working directory from which doxygen will run'. Below this, there is a text input field containing the path '/Users/dongkuk/Downloads/reactor', which is highlighted with a red border. To the right of the input field is a 'Select...' button. Below the input field, it says 'Step 2: Configure doxygen using the Wizard and/or Expert tab, then switch to the Run tab to generate the documentation'. There are three tabs: 'Wizard' (selected), 'Expert', and 'Run'. On the left side of the 'Wizard' tab, there is a 'Topics' list with the following items: 'Project', 'Mode', 'Output', and 'Diagrams'. On the right side of the 'Wizard' tab, there is a section titled 'Provide some information about the project you are documenting'. This section contains four input fields: 'Project name' (with the value 'My Project'), 'Project synopsis', 'Project version or id', and 'Project logo' (with a 'Select...' button). Below this section, there is a section titled 'Specify the directory to scan for source code'. This section contains two input fields: 'Source code directory' (with a 'Select...' button) and a checkbox labeled 'Scan recursively'. Below this section, there is a section titled 'Specify the directory where doxygen should put the generated documentation'. This section contains one input field: 'Destination directory' (with a 'Select...' button'). At the bottom of the window, there are two buttons: 'Previous' and 'Next'.

Doxygen GUI frontend

Step 1: Specify the working directory from which doxygen will run

/Users/dongkuk/Downloads/reactor Select...

Step 2: Configure doxygen using the Wizard and/or Expert tab, then switch to the Run tab to generate the documentation

Wizard Expert Run

Topics

- Project
- Mode
- Output
- Diagrams

Provide some information about the project you are documenting

Project name: My Project

Project synopsis:

Project version or id:

Project logo: Select...

Specify the directory to scan for source code

Source code directory: Select...

☐ Scan recursively

Specify the directory where doxygen should put the generated documentation

Destination directory: Select...

Previous Next

소스 파일이 있는 폴더를 지정합니다.

The screenshot shows the 'Doxygen GUI frontend' window. It has a title bar with standard macOS window controls. The main content area is divided into two steps:

Step 1: Specify the working directory from which doxygen will run

A text field contains the path `/Users/dongkuk/Downloads/reactor`, and a 'Select...' button is to its right.

Step 2: Configure doxygen using the Wizard and/or Expert tab, then switch to the Run tab to generate the documentation

Below this, there are three tabs: 'Wizard' (selected), 'Expert', and 'Run'. The 'Wizard' tab is active, showing a 'Topics' list on the left and configuration fields on the right.

Topics:

- Project
- Mode
- Output
- Diagrams

Provide some information about the project you are documenting

Project name: `My Project`

Project synopsis: (empty text field)

Project version or id: (empty text field)

Project logo: (empty text field) [Select...]

Specify the directory to scan for source code

Source code directory: `s/dongkuk/Downloads/reactor/src` [Select...]

☒ Scan recursively

Specify the directory where doxygen should put the generated documentation

Destination directory: (empty text field) [Select...]

At the bottom, there are 'Previous' and 'Next' buttons.

모든 하위 폴더를 재귀적으로 스캔합니다.

The screenshot shows the 'Doxygen GUI frontend' window. It has a title bar with standard macOS window controls (red, yellow, green buttons). The main content area is divided into two steps:

Step 1: Specify the working directory from which doxygen will run

A text field contains the path `/Users/dongkuk/Downloads/reactor`, and a 'Select...' button is to its right.

Step 2: Configure doxygen using the Wizard and/or Expert tab, then switch to the Run tab to generate the documentation

Below this, there are three tabs: 'Wizard' (selected), 'Expert', and 'Run'. The 'Wizard' tab is active, showing a 'Topics' list on the left and configuration fields on the right.

Topics List:

- Project
- Mode
- Output
- Diagrams

Configuration Fields:

Provide some information about the project you are documenting

- Project name: `My Project`
- Project synopsis: (empty text field)
- Project version or id: (empty text field)
- Project logo: 'Select...' button and an empty image placeholder.

Specify the directory to scan for source code

- Source code directory: `s/dongkuk/Downloads/reactor/src` (highlighted with a blue box) and a 'Select...' button.
- ☒ Scan recursively (highlighted with a red box)

Specify the directory where doxygen should put the generated documentation

- Destination directory: (empty text field) and a 'Select...' button.

At the bottom, there are 'Previous' and 'Next' buttons.

Doxygen이 문서를 생성할 폴더를 지정합니다.

The screenshot shows the 'Doxygen GUI frontend +' window. It has a title bar with standard macOS window controls. The main content area is divided into two steps:

Step 1: Specify the working directory from which doxygen will run

A text field contains the path `/Users/dongkuk/Downloads/reactor`, followed by a `Select...` button.

Step 2: Configure doxygen using the Wizard and/or Expert tab, then switch to the Run tab to generate the documentation

Below this, there are three tabs: `Wizard` (selected), `Expert`, and `Run`.

Wizard Tab:

- Topics:** A list on the left includes `Project`, `Mode`, `Output`, and `Diagrams`.
- Provide some information about the project you are documenting:**
 - `Project name:` `My Project`
 - `Project synopsis:` (empty text field)
 - `Project version or id:` (empty text field)
 - `Project logo:` `Select...` button and a placeholder image box.
- Specify the directory to scan for source code:**
 - `Source code directory:` `s/dongkuk/Downloads/reactor/src` with a `Select...` button.
 - ☒ `Scan recursively`
- Specify the directory where doxygen should put the generated documentation:** (This section is highlighted with a red rectangle)
 - `Destination directory:` `ongkuk/Downloads/reactor/untitled` with a `Select...` button.

At the bottom, there are `Previous` and `Next` buttons.

프로젝트 이름과 버전을 지정합니다.

The screenshot shows the 'Doxygen GUI frontend +' window. It has two main steps:

Step 1: Specify the working directory from which doxygen will run

The working directory is set to `/Users/dongkuk/Downloads/reactor`. A 'Select...' button is available.

Step 2: Configure doxygen using the Wizard and/or Expert tab, then switch to the Run tab to generate the documentation

The 'Wizard' tab is selected. On the left, a 'Topics' list includes 'Project', 'Mode', 'Output', and 'Diagrams'. The main area is titled 'Provide some information about the project you are documenting' and contains the following fields:

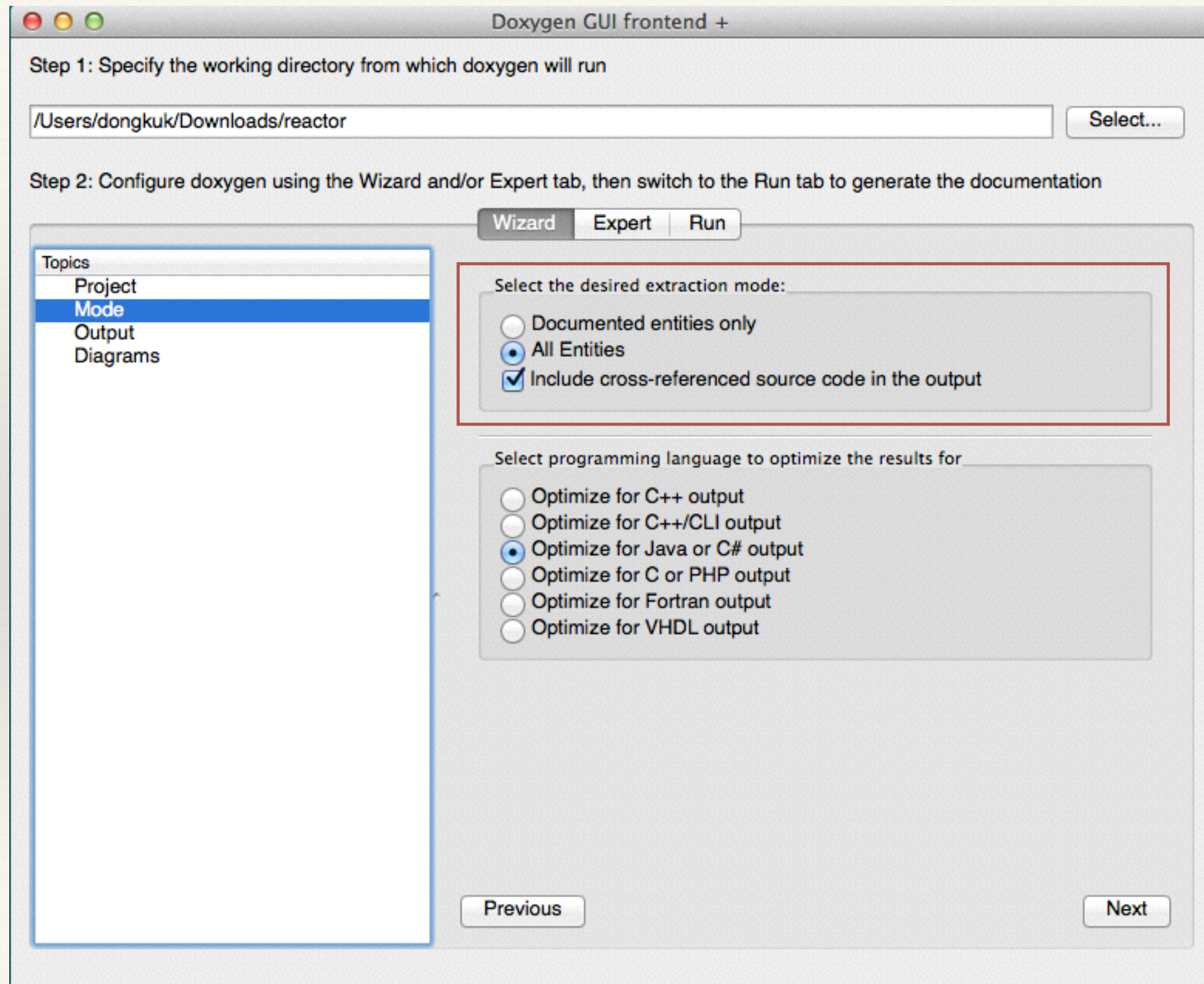
- Project name:** `Reactor`
- Project synopsis:** (empty text box)
- Project version or id:** `1.0.0` (highlighted with a blue border)
- Project logo:** 'Select...' button and an empty image placeholder.

Below this section, there are two more configuration areas:

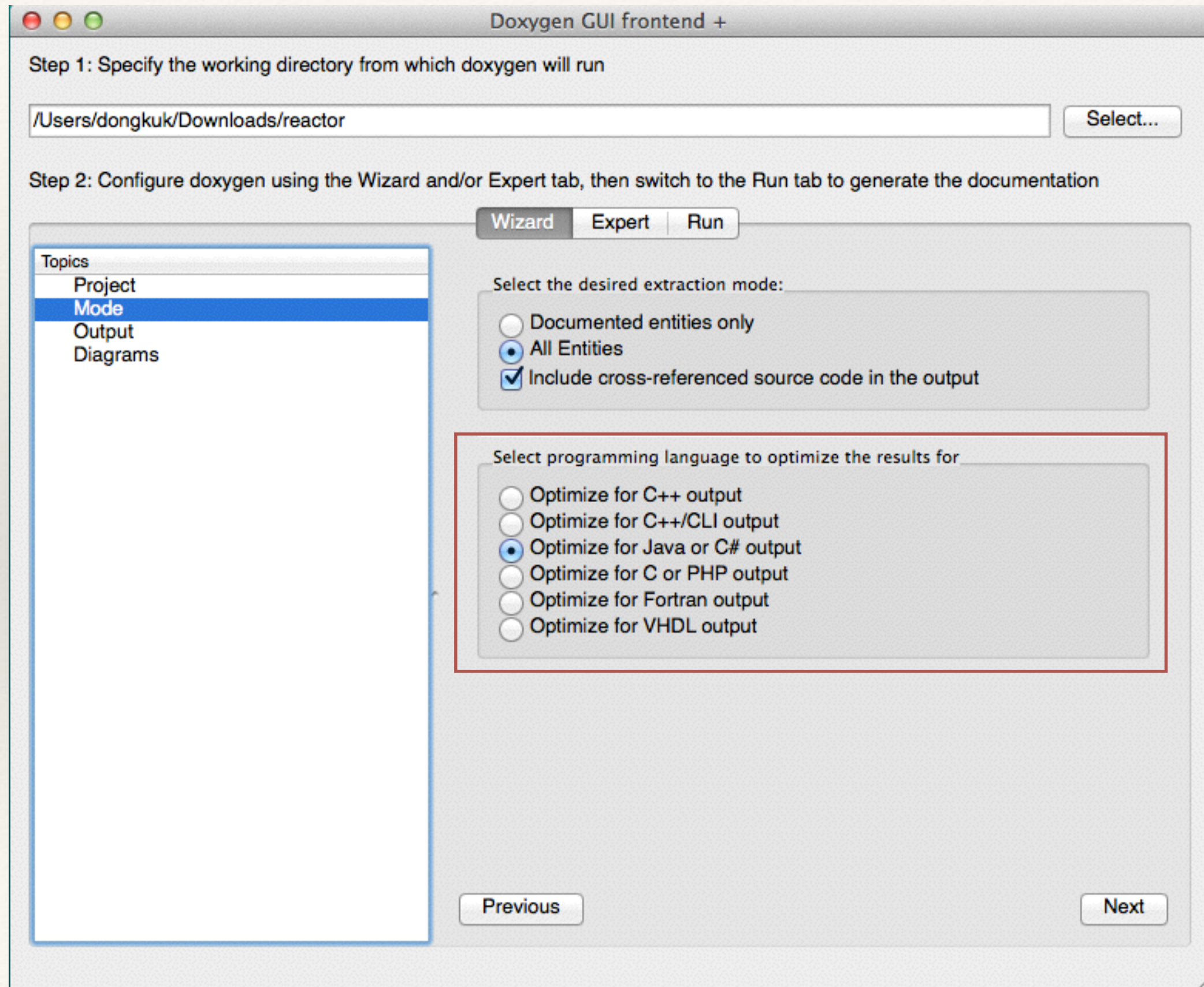
- Specify the directory to scan for source code**
 - Source code directory:** `s/dongkuk/Downloads/reactor/src` with a 'Select...' button.
 - ☒ **Scan recursively**
- Specify the directory where doxygen should put the generated documentation**
 - Destination directory:** `ongkuk/Downloads/reactor/untitled` with a 'Select...' button.

At the bottom, there are 'Previous' and 'Next' buttons.

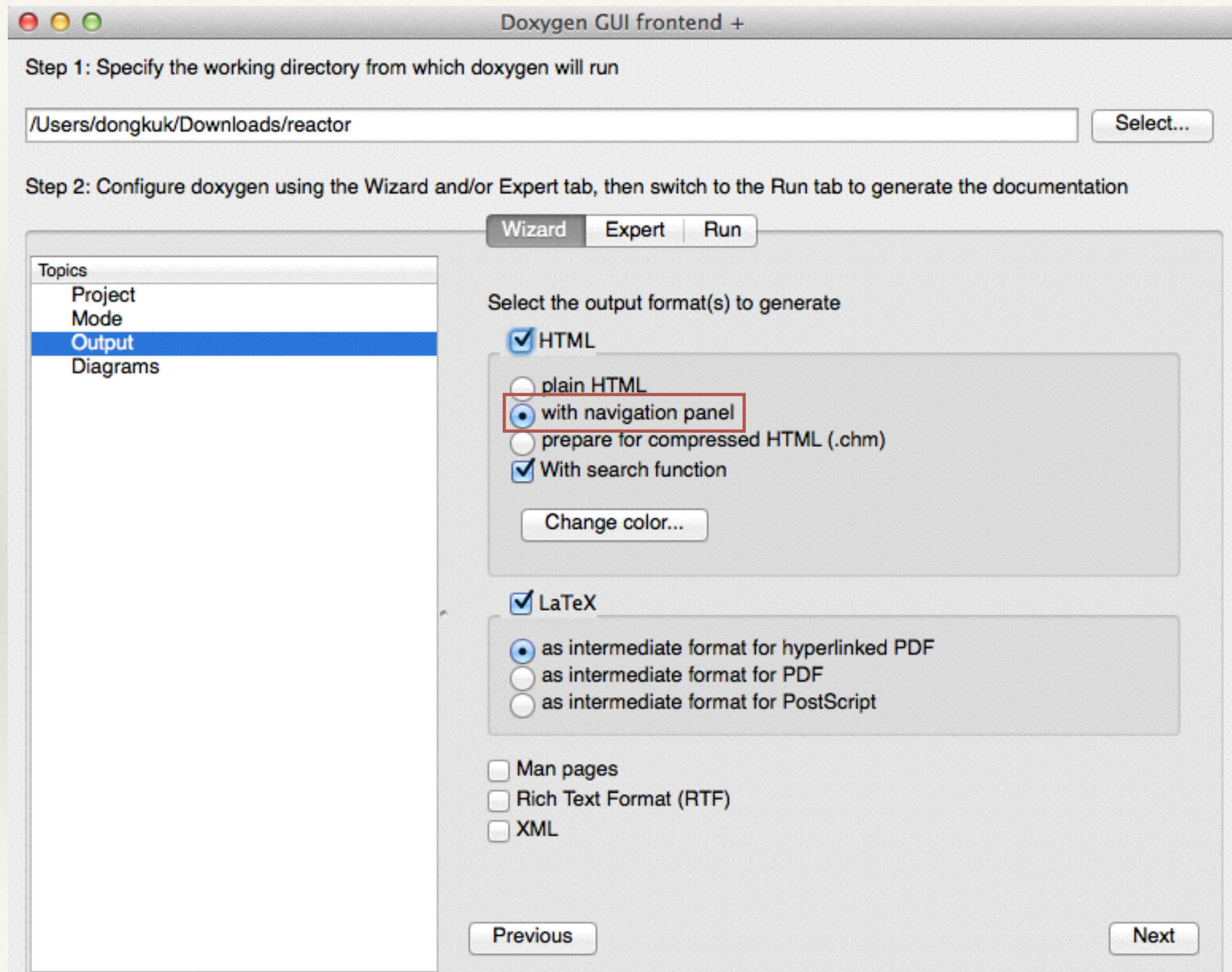
각 함수마다 사용한 함수로의 링크를 생성합니다.



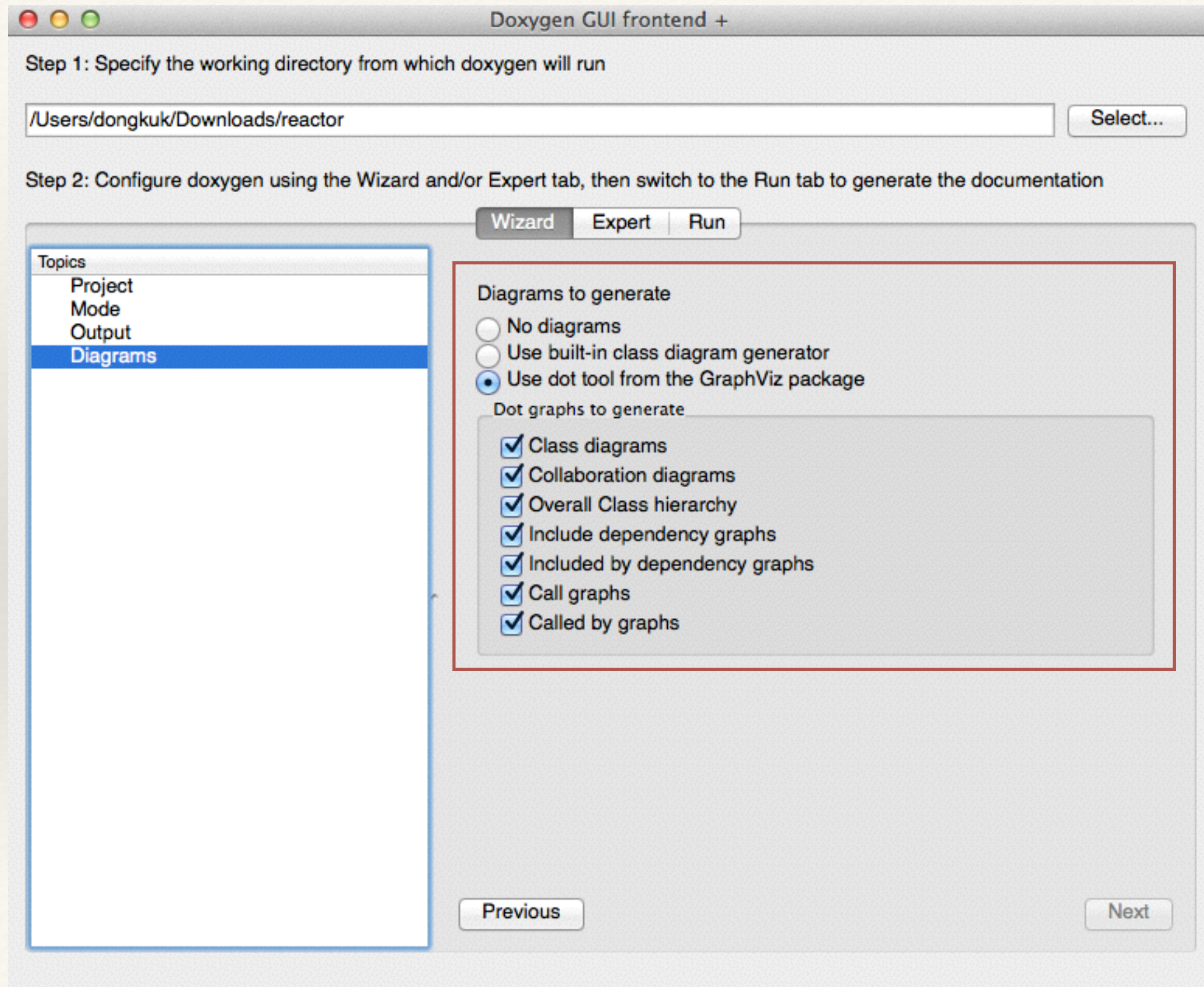
자바를 선택합니다.



문서 왼쪽에 탐색 트리를 보여줍니다.



소스 간의 관계를 **GraphViz**로 표현해줍니다.



출력 결과에 쓰여질 언어를 한국어로 선택합니다.

Doxygen GUI frontend +

Step 1: Specify the working directory from which doxygen will run

/Users/dongkuk/Downloads/reactor Select...

Step 2: Configure doxygen using the Wizard and/or Expert tab, then switch to the Run tab to generate the documentation

Wizard **Expert** Run

Topics

- Project
- Build
- Messages
- Input
- Source Browser
- Index
- HTML
- LaTeX
- RTF
- Man
- XML
- Docbook
- AutoGen
- PerlMod

PROJECT_LOGO

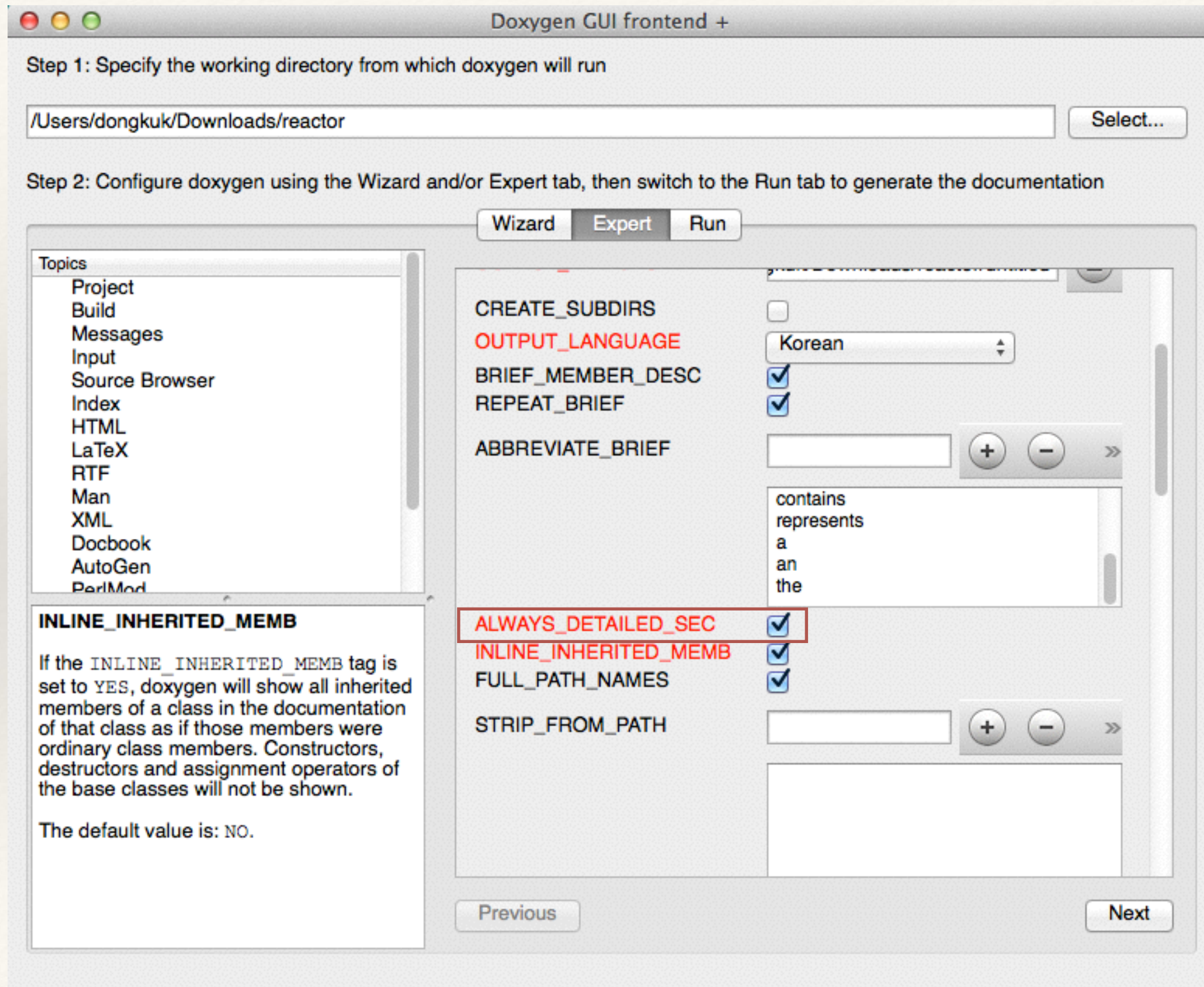
With the `PROJECT_LOGO` tag one can specify an logo or icon that is included in the documentation. The maximum height of the logo should not exceed 55 pixels and the maximum width should not exceed 200 pixels. Doxygen will copy the logo to the output directory.

DOXYFILE_ENCODING	UTF-8
PROJECT_NAME	Reactor
PROJECT_NUMBER	1.0.0
PROJECT_BRIEF	
PROJECT_LOGO	<input type="text"/>
OUTPUT_DIRECTORY	/Users/dongkuk/Downloads/reactor/untitled
CREATE_SUBDIRS	<input type="checkbox"/>
OUTPUT_LANGUAGE	Korean
BRIEF_MEMBER_DESC	<input checked="" type="checkbox"/>
REPEAT_BRIEF	<input checked="" type="checkbox"/>
ABBREVIATE_BRIEF	<input type="text"/> + - >>

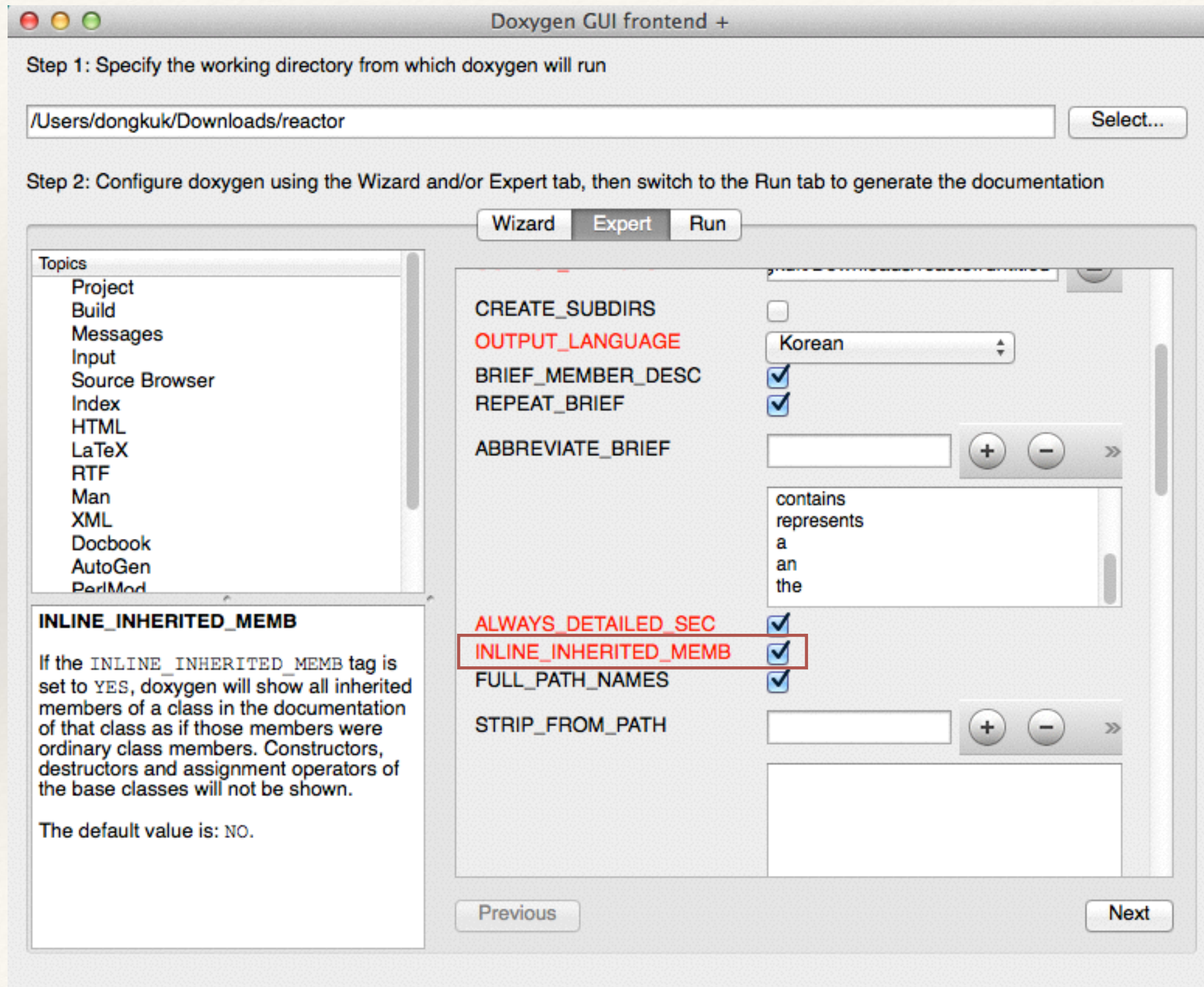
The \$name class
The \$name widget
The \$name file
is
provides

Previous Next

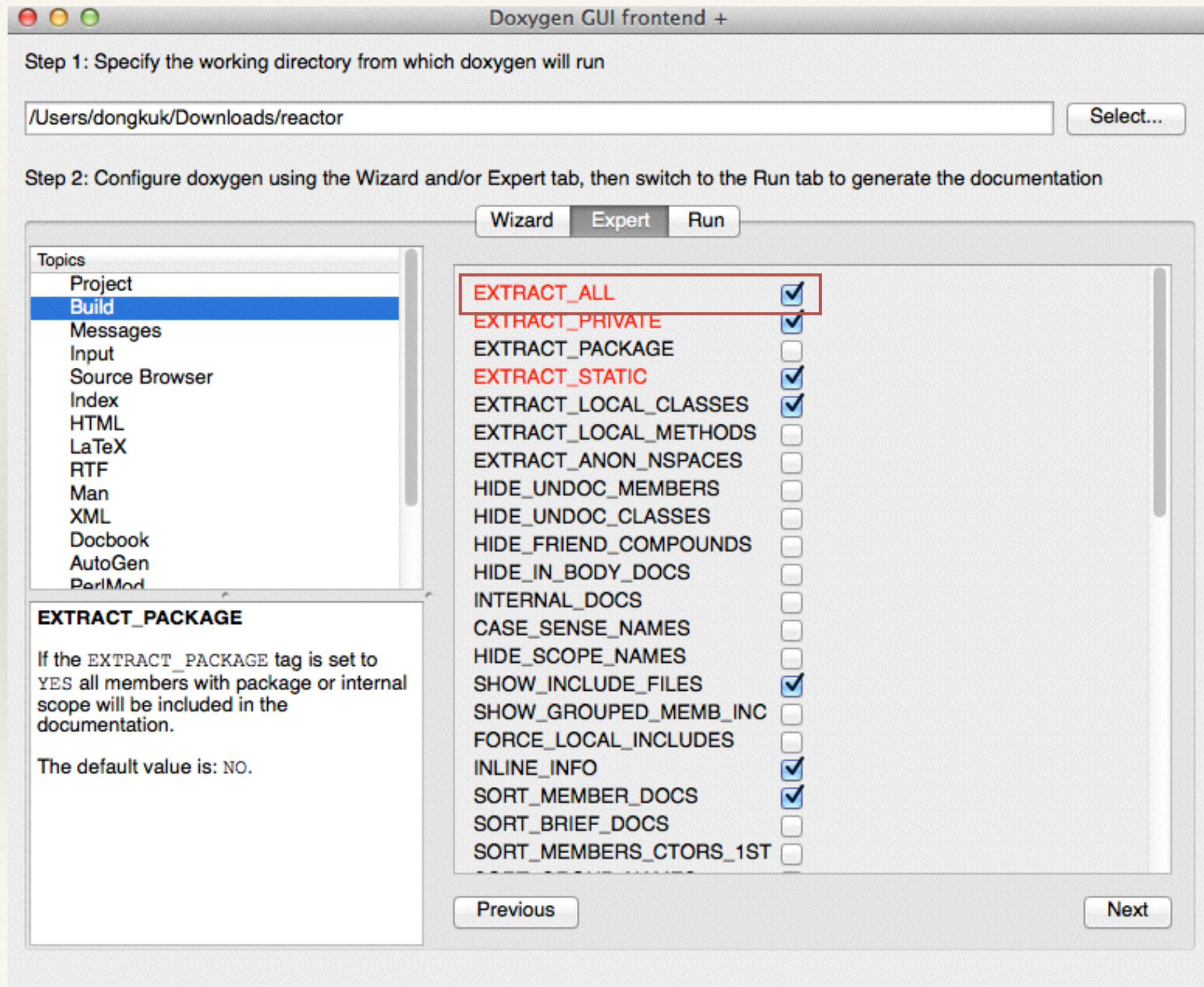
항상 상세 정보를 보여줍니다.



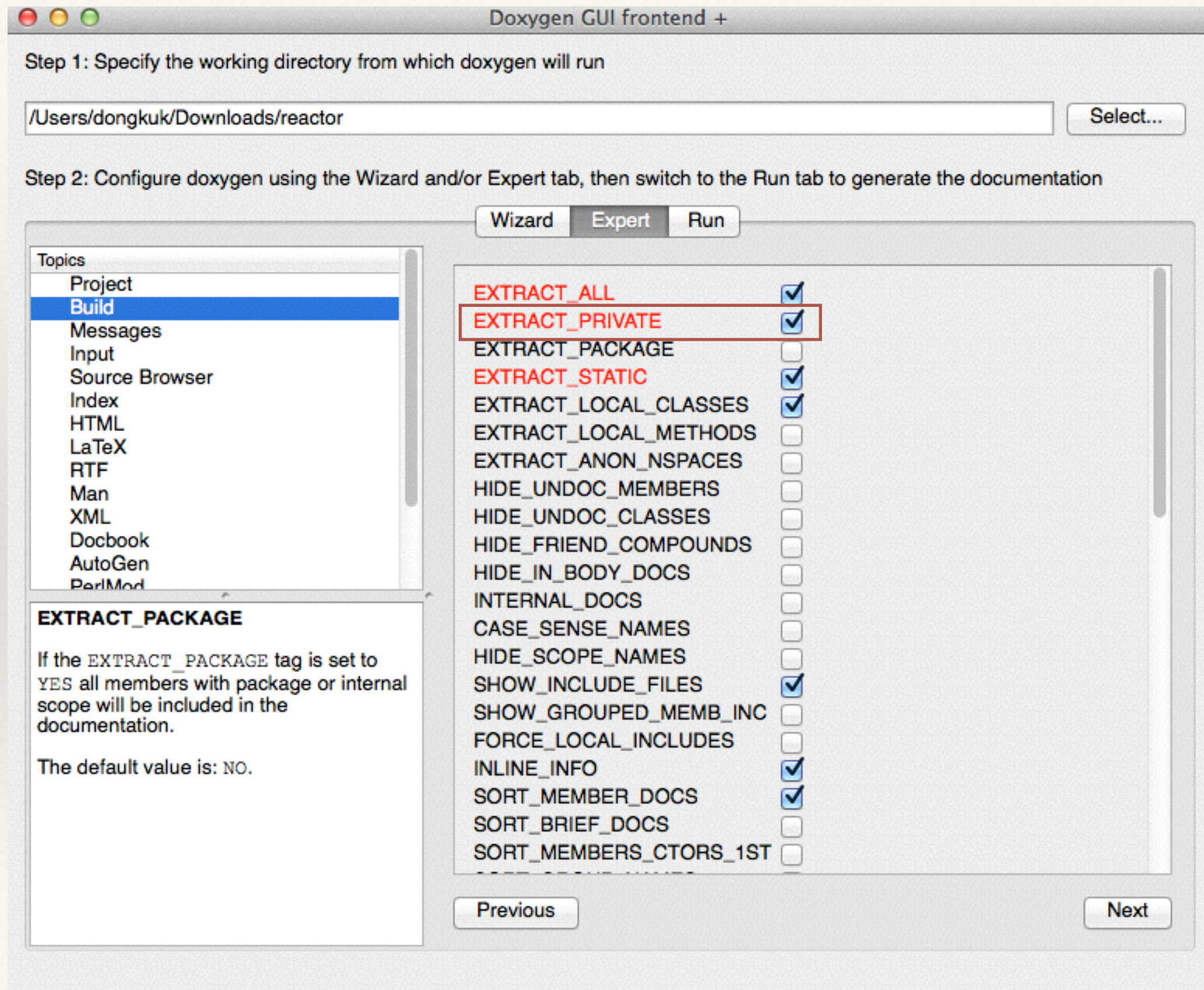
소멸자와 상속자를 제외한 상속된 모든 멤버를 보여줍니다.



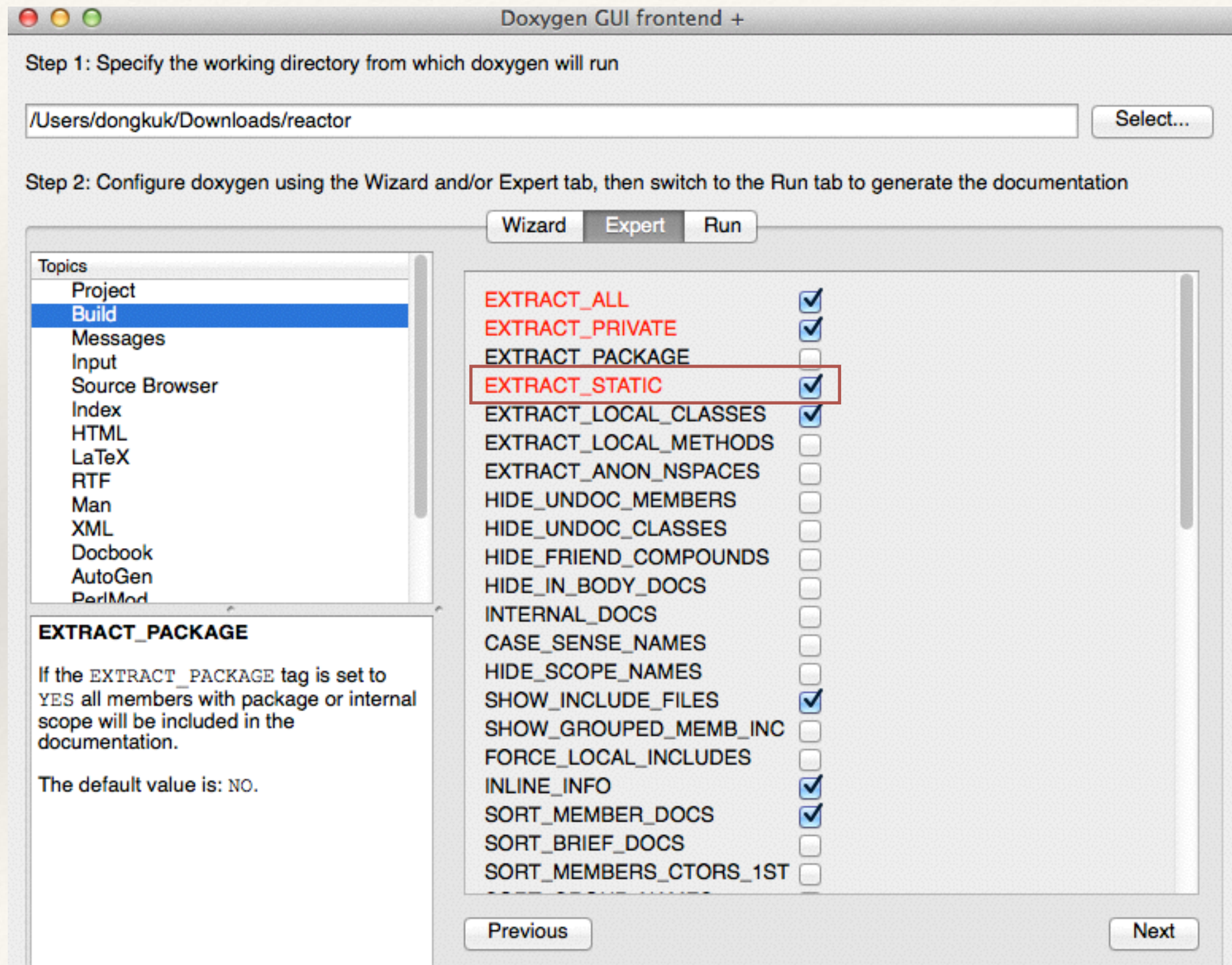
소스코드의 모든 요소가 문서화 대상이 됩니다.



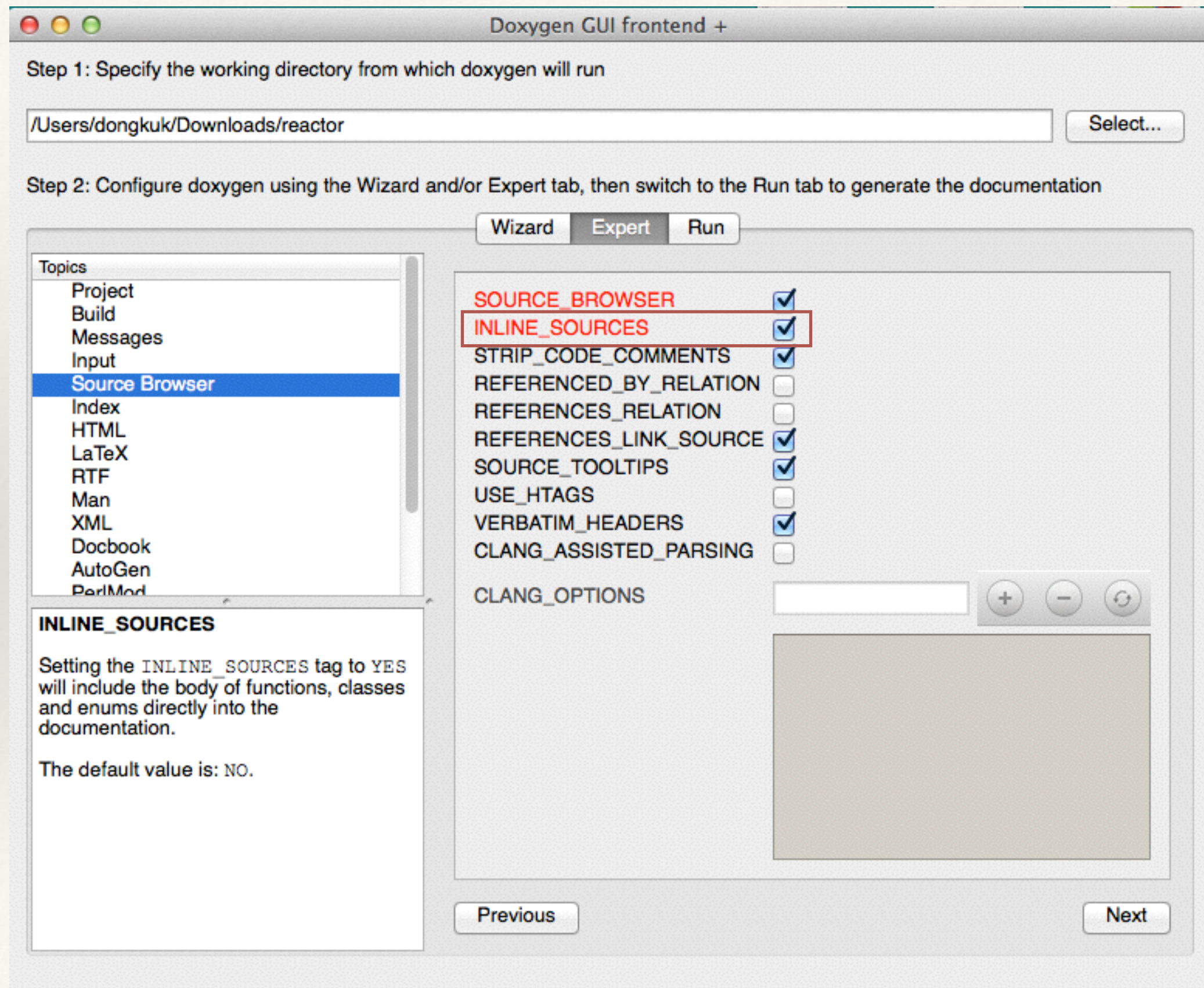
클래스 내의 모든 **private** 멤버가 문서화 대상이 됩니다.



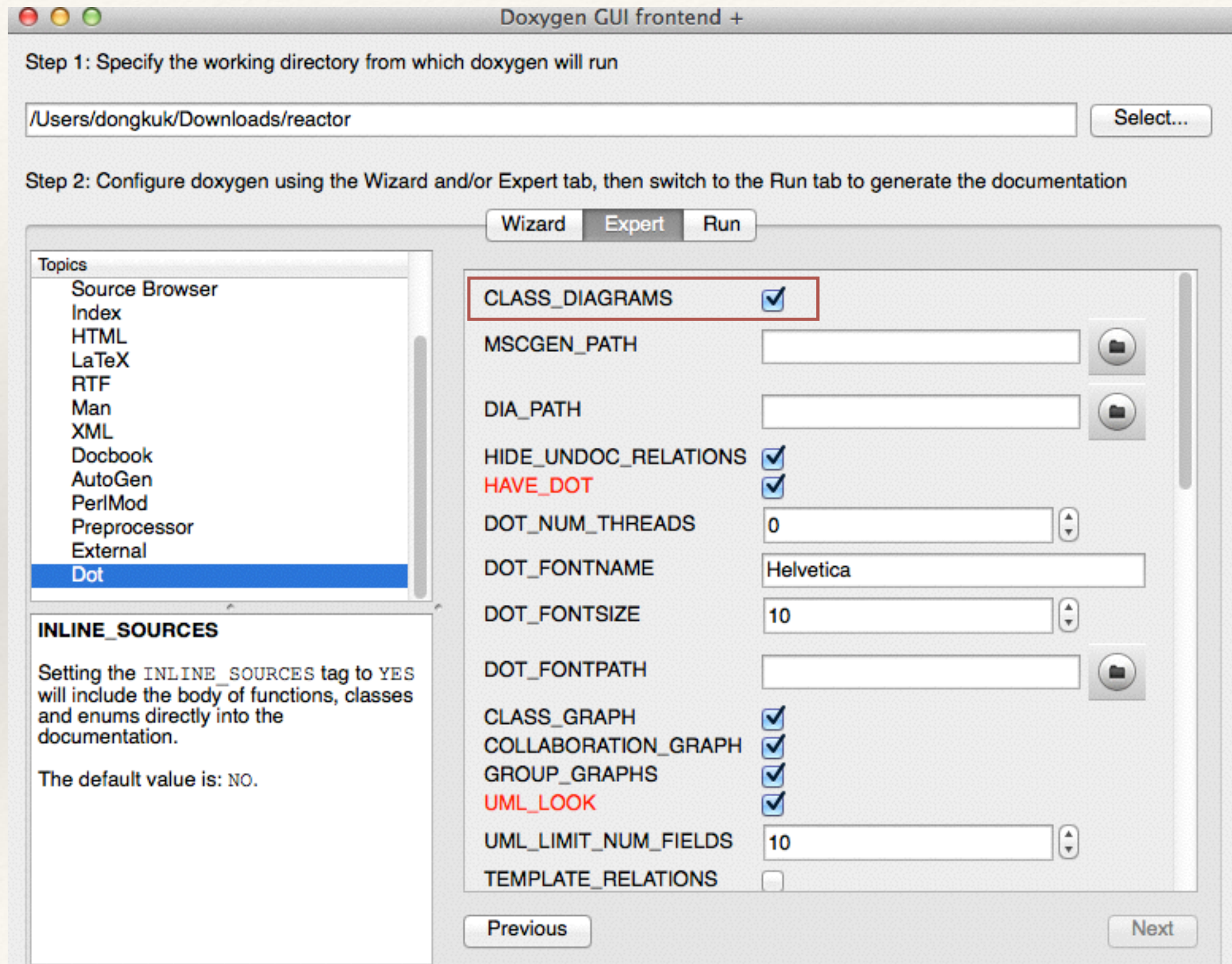
클래스 내의 모든 **static** 멤버가 문서화 대상이 됩니다.



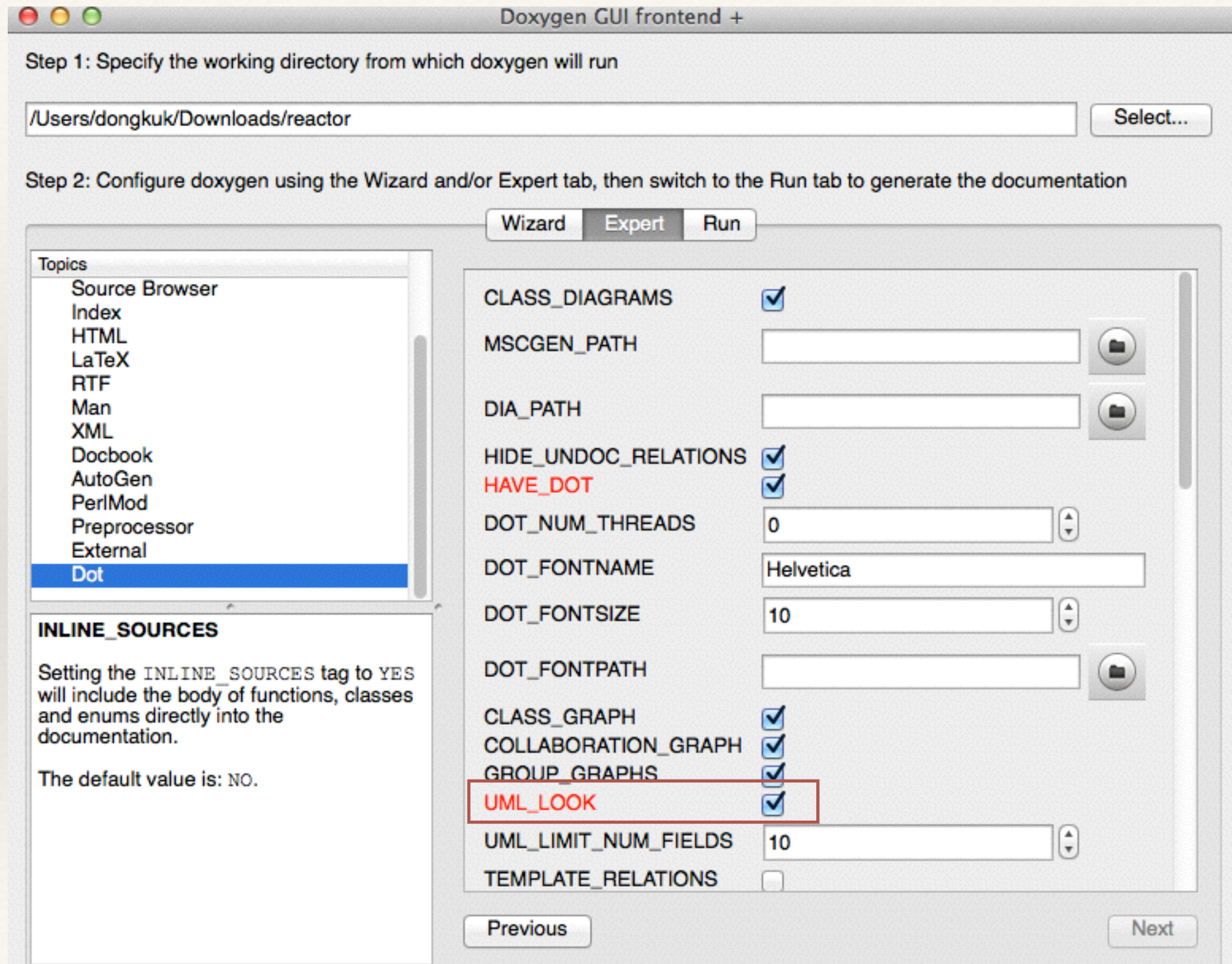
함수 설명시 함수 코드를 보여줍니다.



클래스의 상속구조 다이어그램을 그립니다.

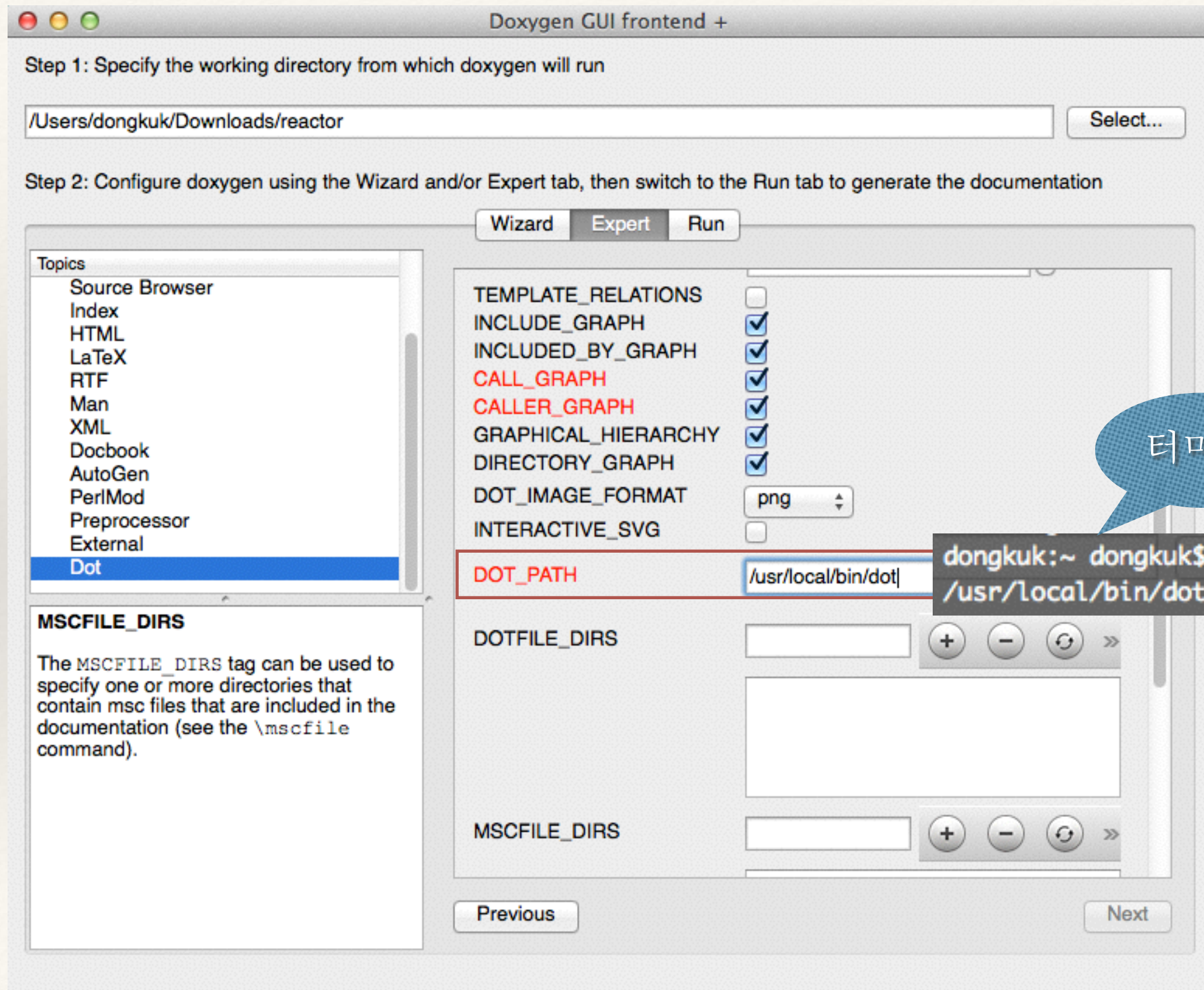


다이어그램을 UML 형식으로 그립니다.



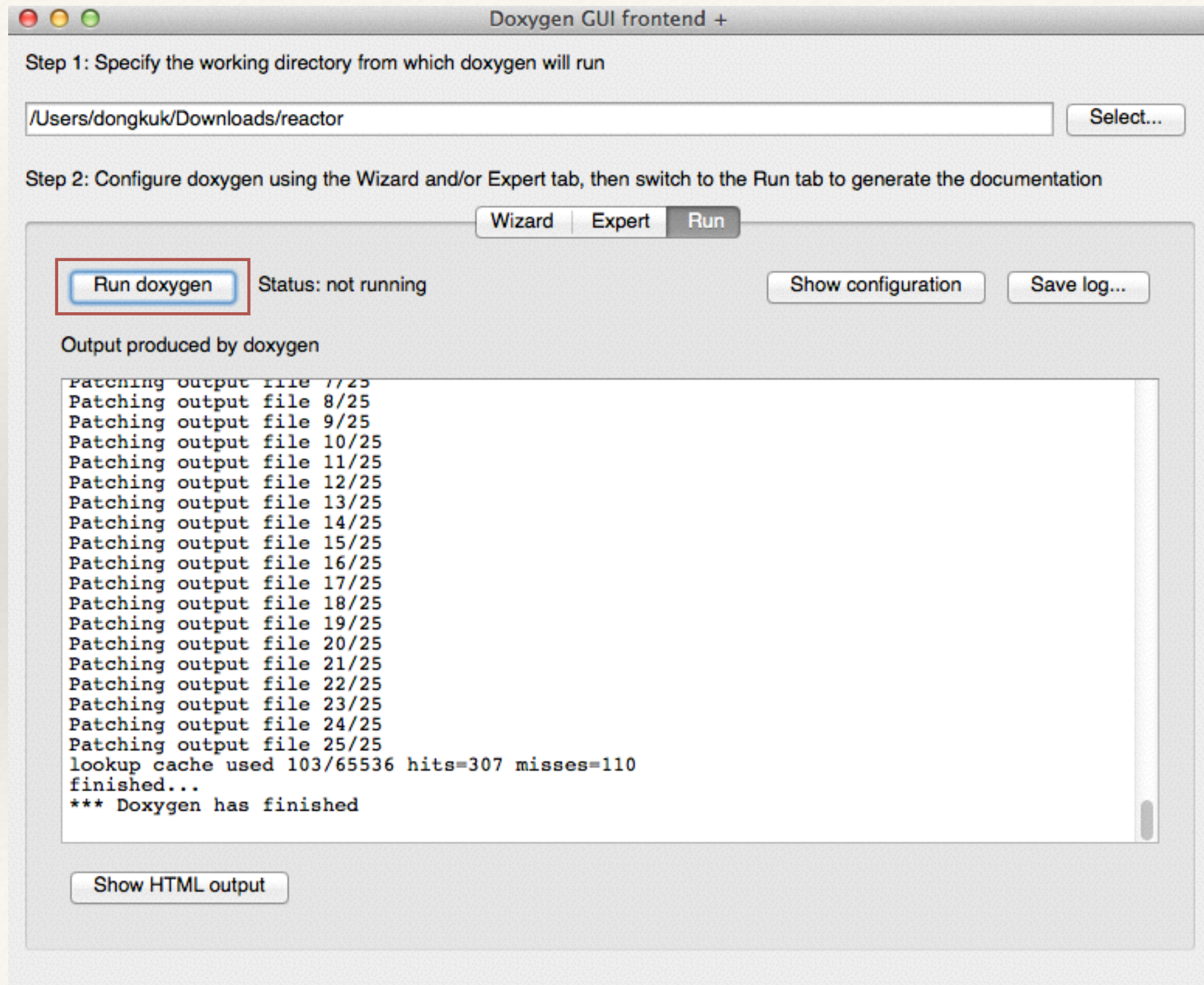
dot_path를 지정해줍니다.

dot 프로그램이란 Graphviz을 의미합니다!

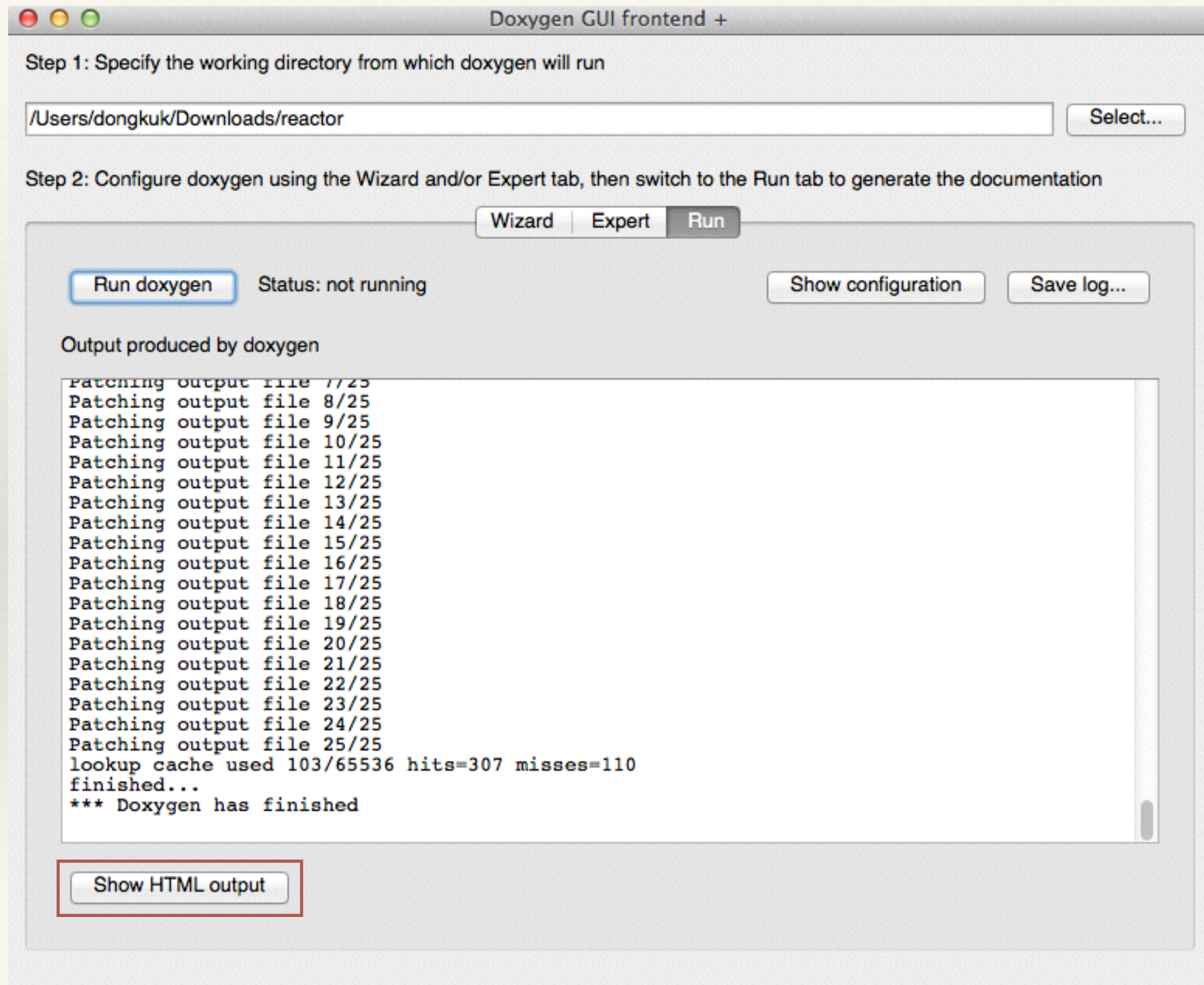


이것으로 **Doxygen** 설정 세팅은 마쳤습니다.
이제 문서를 생성해보아요.

Doxygen을 시작해 문서를 생성합니다.

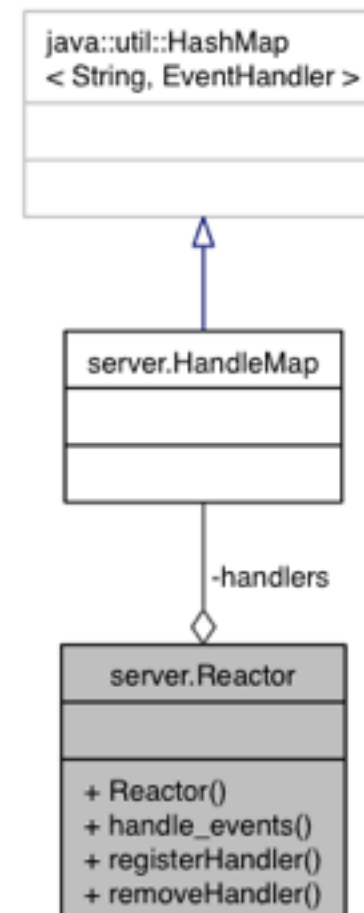


생성된 문서를 열어봅니다.



server.Reactor 클래스 참조

server.Reactor에 대한 협력 다이어그램:



Public 멤버 함수

끝