

Doxygen

Bogdan Kreczmer

ZPCiR ICT PWr

pokój 307 budynek C3

kreczmer@ict.pwr.wroc.pl

Copyright ©2004 Bogdan Kreczmer*

* Niniejszy dokument zawiera materiały do wykładu na temat programowania obiektowego. Jest on udostępniony pod warunkiem wykorzystania wyłącznie do własnych prywatnych potrzeb i może on być kopiowany wyłącznie w całości, razem z niniejszą stroną tytułową.

Projekty wykorzystujące doxygen

Autor projektu: Dimitri van Heesch

Strona projektu: <http://www.doxygen.org> (<http://www.stack.nl/~dimitri/doxygen/>)

Doxygen jest systemem dokumentowania oprogramowania pisanego w C++, C, Java, Objective-C, IDL, C# oraz D.

System ten może generować dokumentację w HTML oraz źródła dla \LaTeX 'a. Dokumentacja może mieć format RTF (MS-Word), PostScript, PDF (z *hyper*-łącznikami), skompresowanego HTML oraz stron podręcznika w systemie UNIX dla programu `man`.

Dokumentacja może być generowana bezpośrednio ze źródeł jak też plików stowarzyszonych z nimi.

Doxygen jest tworzonym dla systemu Linux i Mac OS w oparciu o bibliotekę `Qt`. Oprogramownie to jest przenośne. Może być uruchomione pod system UNIX i jego klonami. Dostępne jest również w postaci binarnej dla system MS Windows.

Projekty wykorzystujące doxygen

Bardziej znane i ciekawsze projekty wykorzystujące do tworzenia dokumentacji **doxygen**:

- **KDE** - dokumentacja różnych bibliotek.
- **Samba**
- **OSCAR** (<http://www.oscar-net.org>) tworzy ogólną architekturę dla systemów robotycznych z ukierunkowaniem na roboty mobilne.
- **XEngine** (<http://xengine.sourceforge.net>) “silnik” dla wizualizacji 3D w czasie rzeczywistym, jest on niezależny od platformy i typu API wykorzystywanego renderowania.

Inne systemy tworzenia dokumentacji

Niekomercyjne:

- AutoDOC
- Autoduck
- Cocoon
- CcDoc
- CppDoc
- Cxref
- cxxwrap
- Cxx2HTML
- C2HTML
- Doc++
- DocClass
- Epydoc
- gtk-doc
- HappyDoc
- HeaderDoc
- HTMLgen
- HyperSQL
- Javadoc
- KDoc
- Natural Docs
- Perceps
- phpDocumentor
- PHPDoc
- ReThree-C++
- RoboDoc
- ScanDoc
- Synopsis
- Tydoc
- VBDOX

Komercyjne:

- DocBuilder
- DocJet
- Doc-o-matic
- ObjectManual
- Together
- CC-Rider
- VBXC

Doxygen - główne komponenty

doxygen - program generujący dokumentację na podstawie utworzonego wcześniej pliku konfiguracyjnego i przeglądanych plików (źródeł programów i nie tylko).

doxytag - program pomocniczy pozwalający integrować zewnętrzną dokumentację (do której **doxygen** nie ma bezpośredniego dostępu) z dokumentacją tworzoną przez **doxygen**.

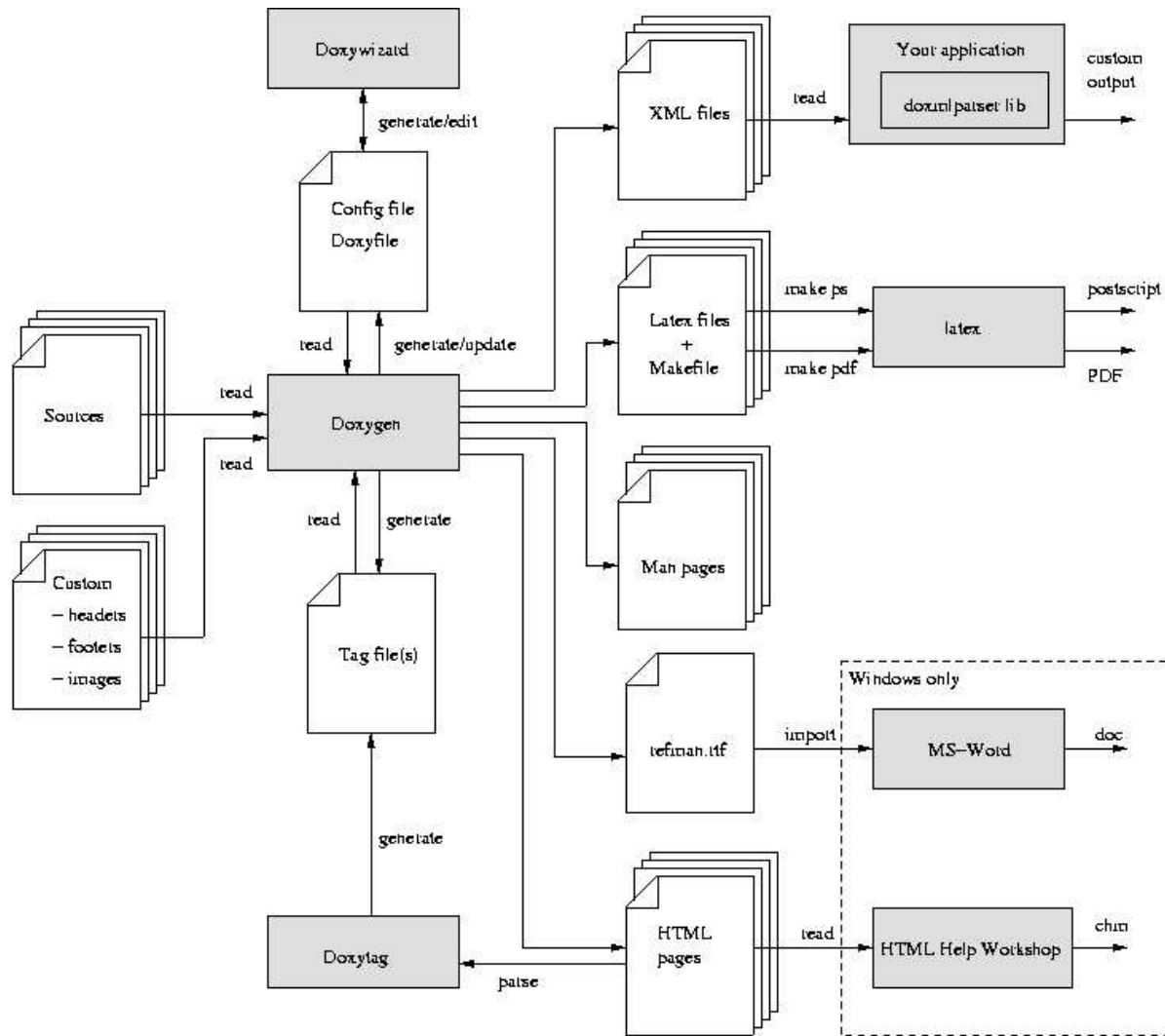
doxywizard - graficzna aplikacja ułatwiająca tworzenie pliku konfiguracyjnego dla dokumentacji danego projektu.

Komponenty stowarzyszone:

graphviz - oprogramowanie wykorzystywane do tworzenia rysunków grafów.

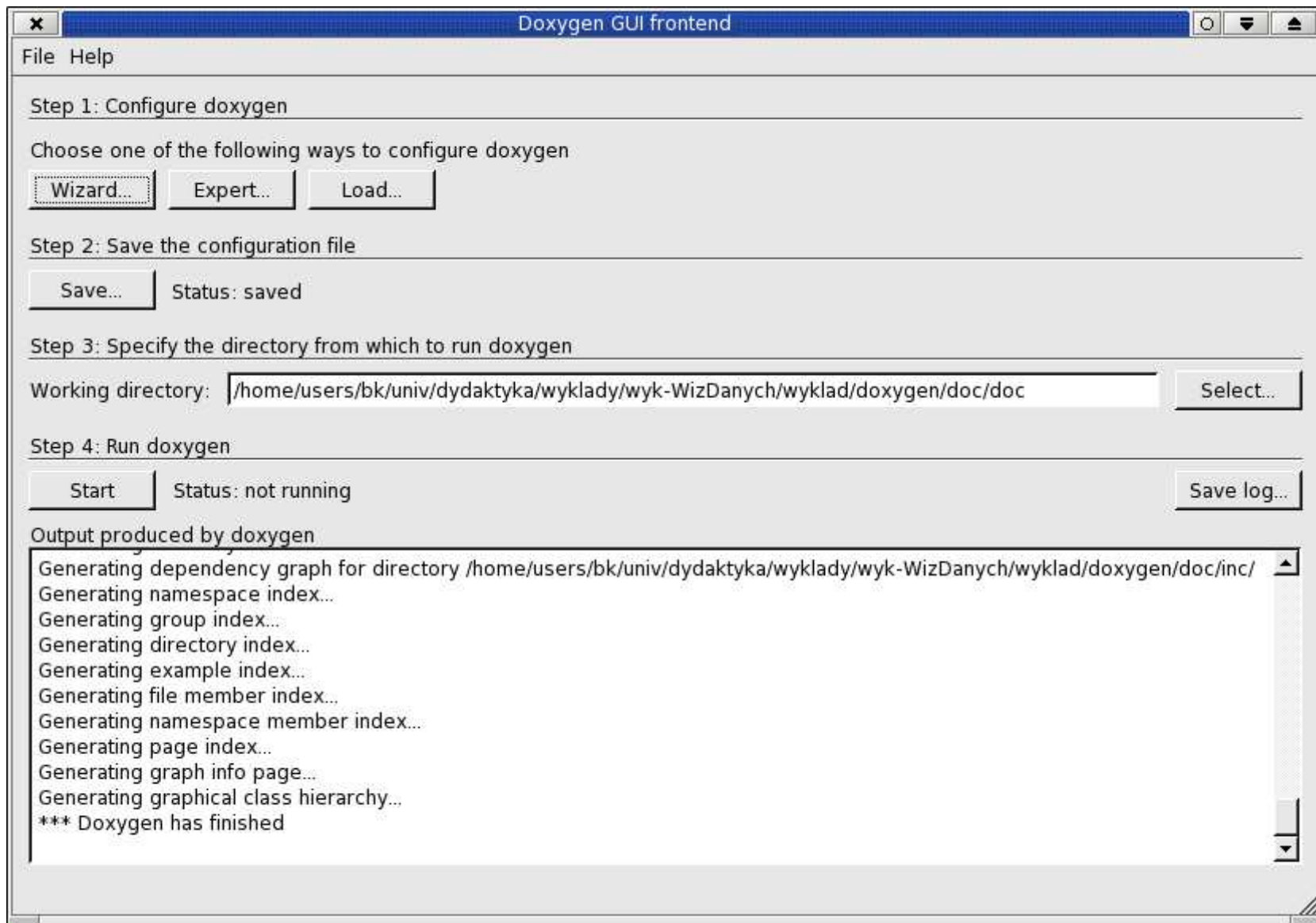
Strona projektu: <http://www.research.att.com/sw/tools/graphviz/>

Doxygen - generacja dokumentacji



Przepływ informacji w Doxygen.

Doxywizard - główne okienko



Rysunek 1: Okienko główne aplikacji doxywizard.

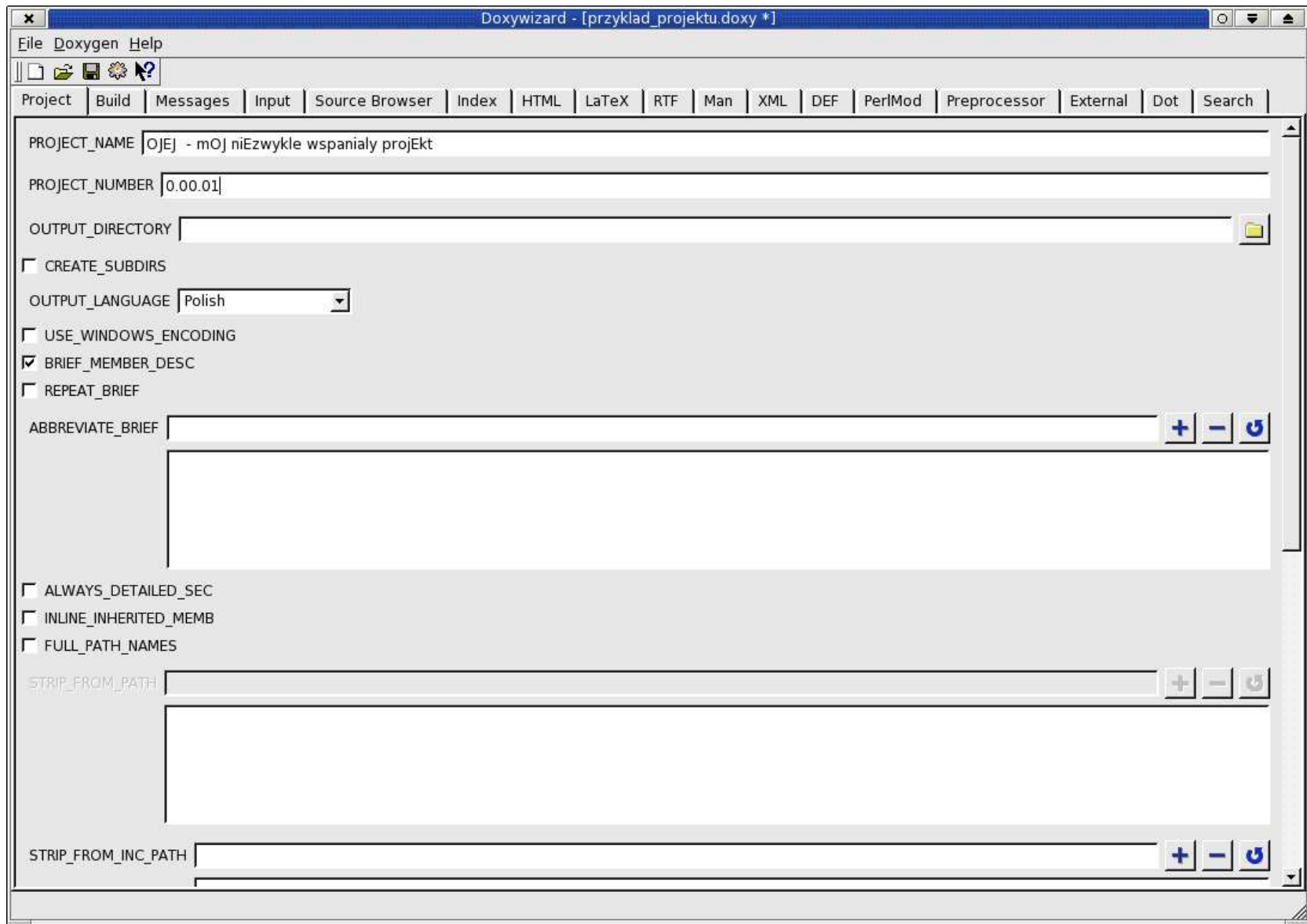
Doxygen - Makefile

```
__start_doxy__:  
    doxygen bec.doxy 2> make.log  
    less make.log
```

Przekierowania wyjścia *standard error* pozwala “odcedzić” komunikaty o błędach od pozostałych kontrolnych komunikatów.

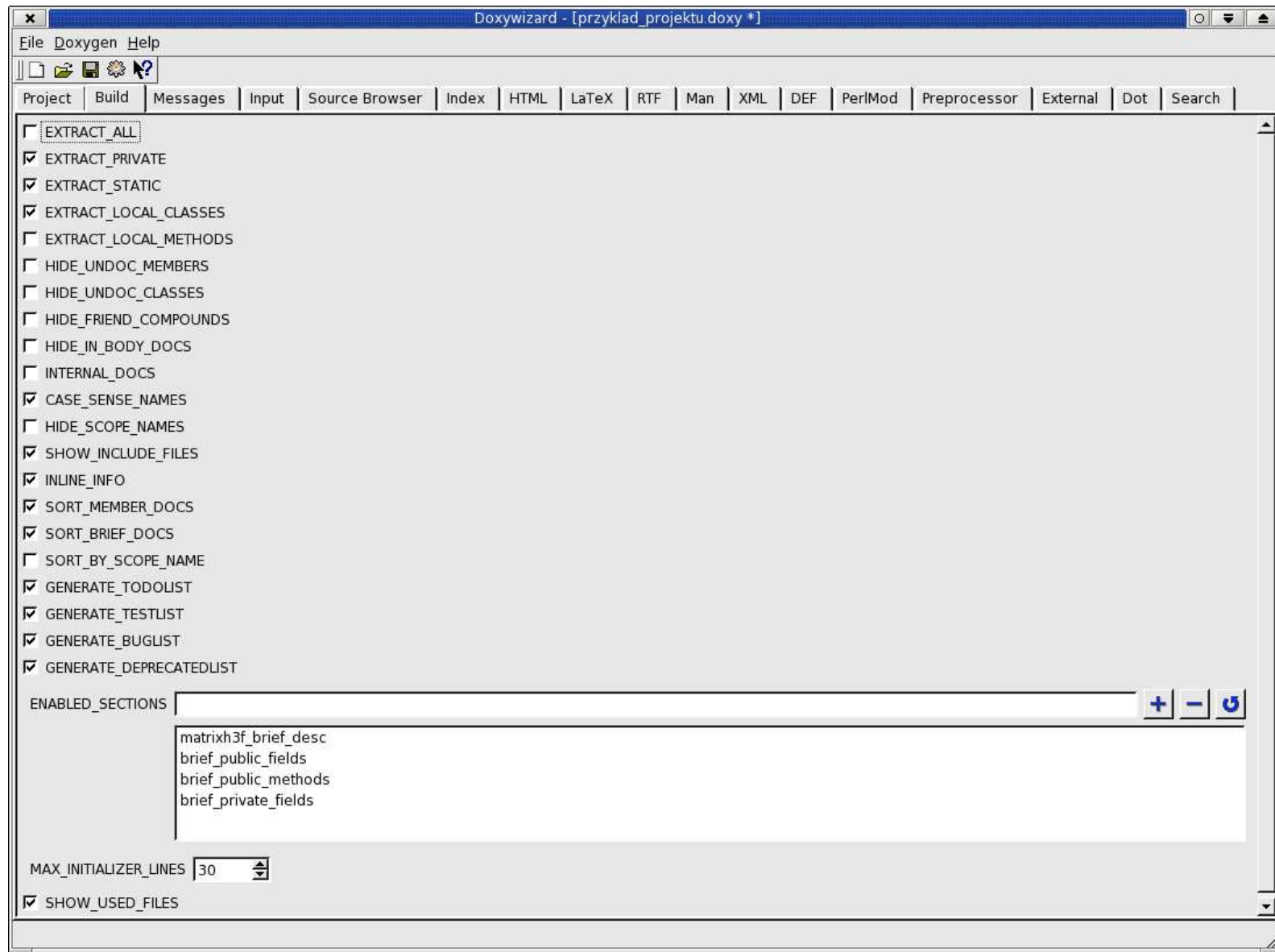
Ze względu na bardzo dużą ilość przetwarzanych plików liczba pozostałych komunikatów jest bardzo duża. Selekcja ich jest więc bardzo wskazana.

Doxywizard - tworzenie pliku konfiguracyjnego (cz. 1)



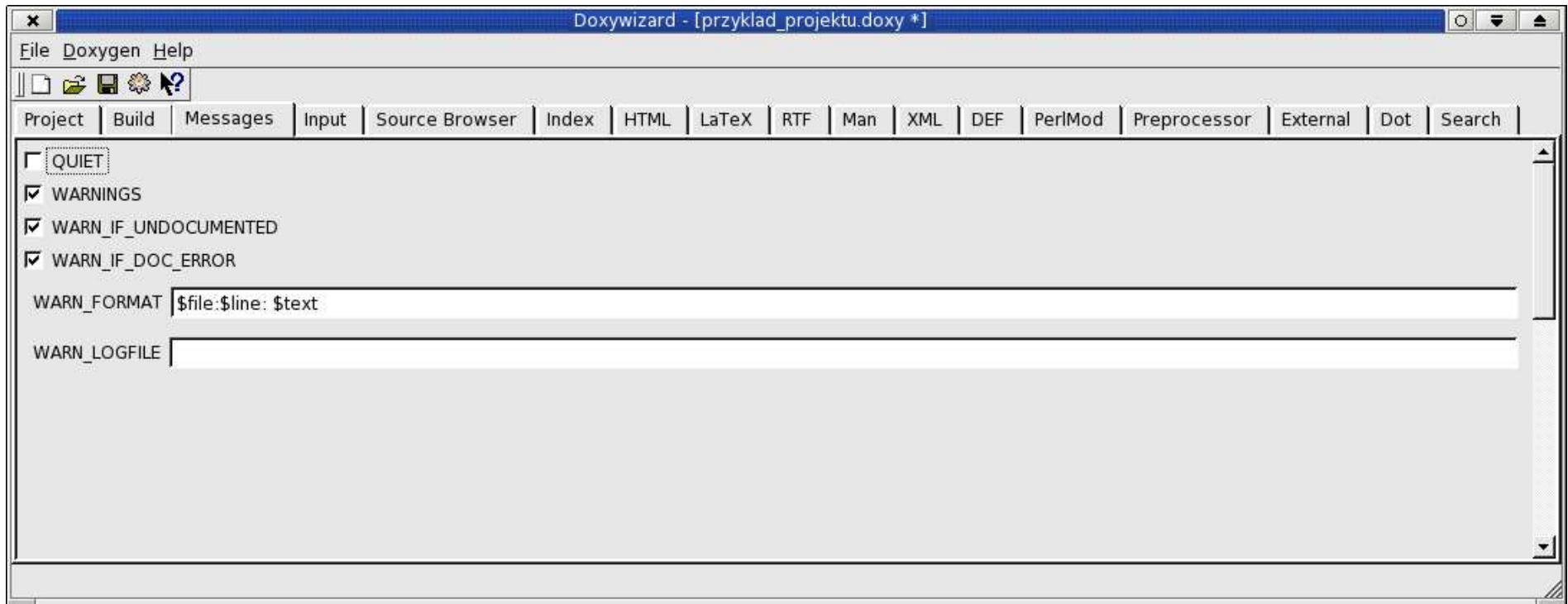
Rysunek 2: Podstawowe opcje.

Doxywizard - tworzenie pliku konfiguracyjnego (cz. 2)



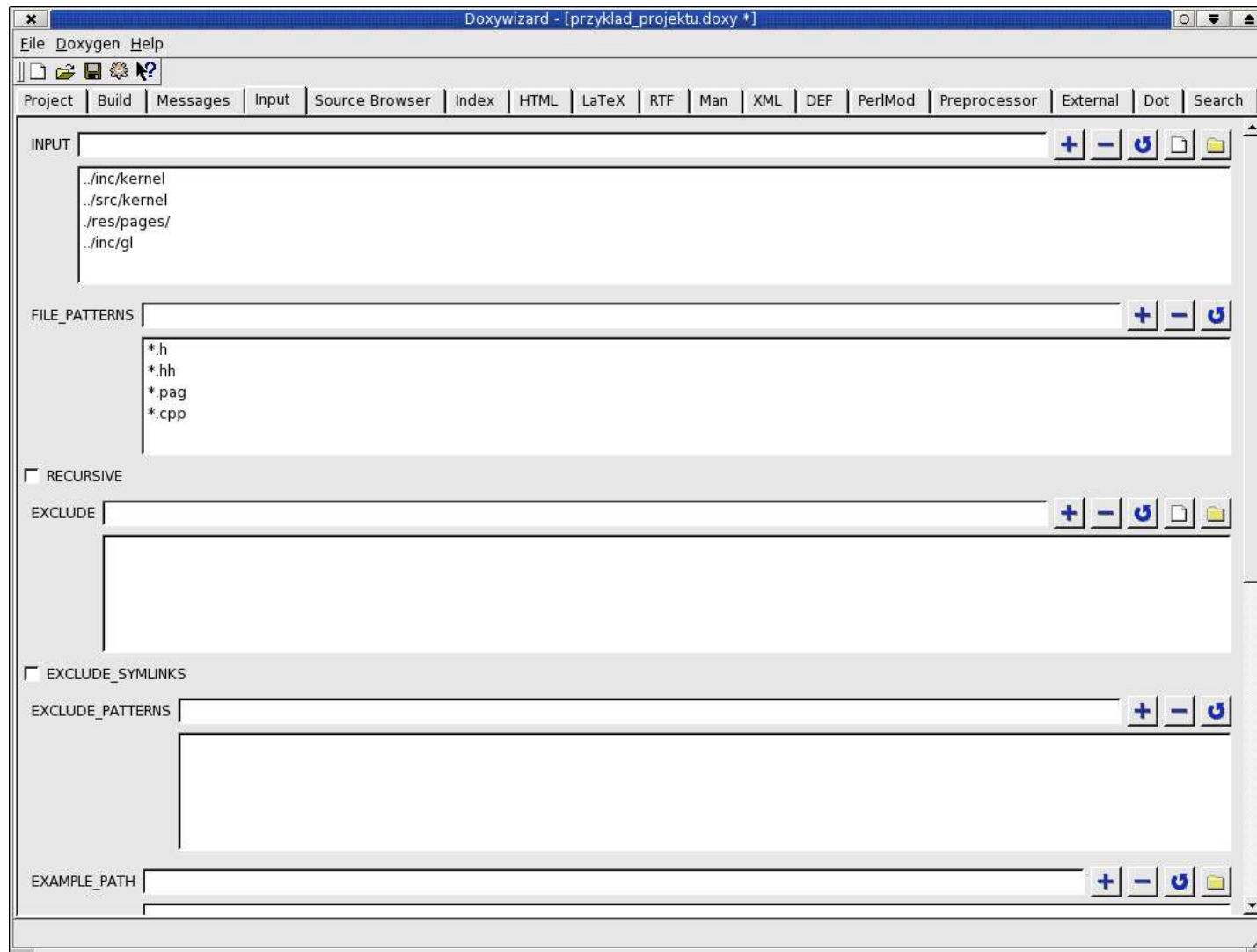
Rysunek 3: Opcje sposobu tworzenia dokumentacji.

Doxywizard - tworzenie pliku konfiguracyjnego (cz. 3)



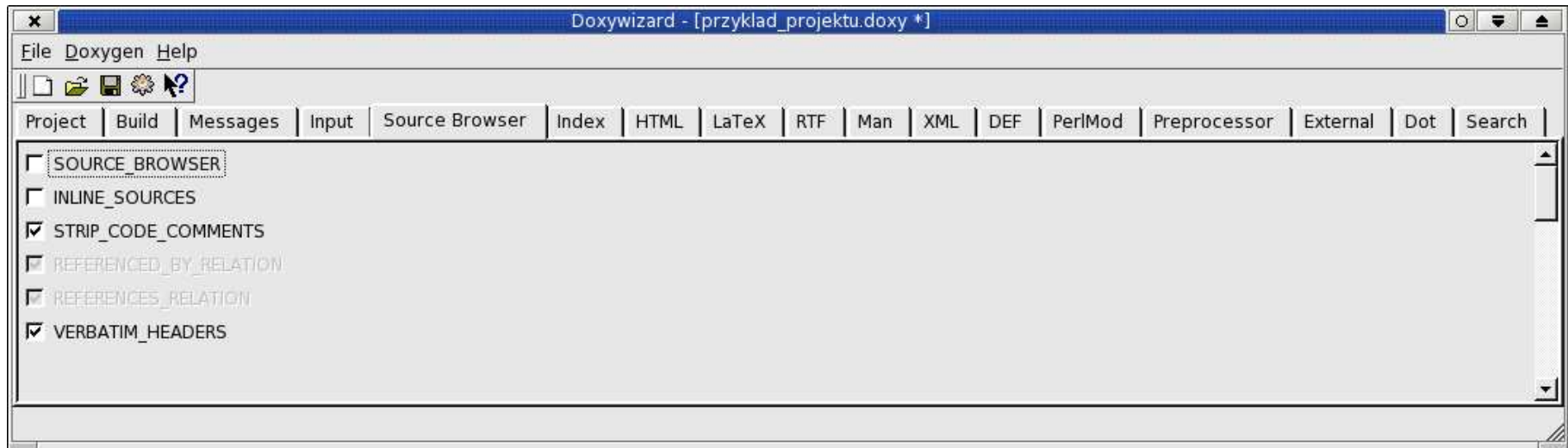
Rysunek 4: Opcje komunikatów.

Doxywizard - tworzenie pliku konfiguracyjnego (cz. 4)



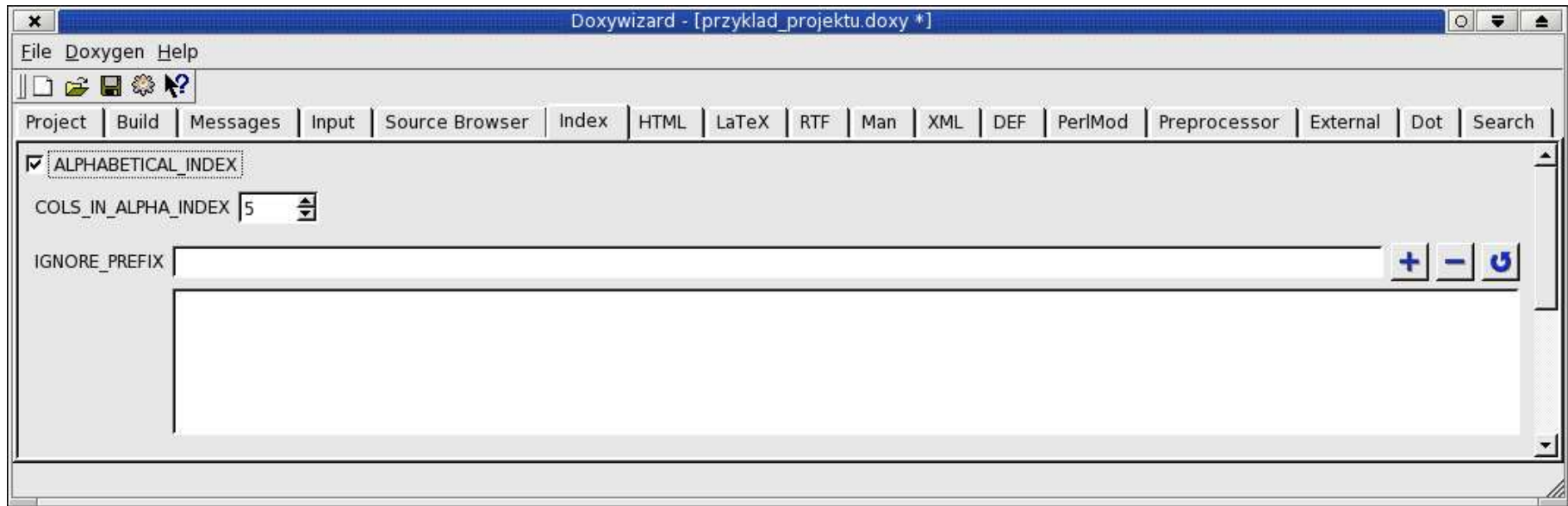
Rysunek 5: Opcje ścieżek do źródeł, przykładów, obrazków oraz wzorców nazw.

Doxywizard - tworzenie pliku konfiguracyjnego (cz. 5)



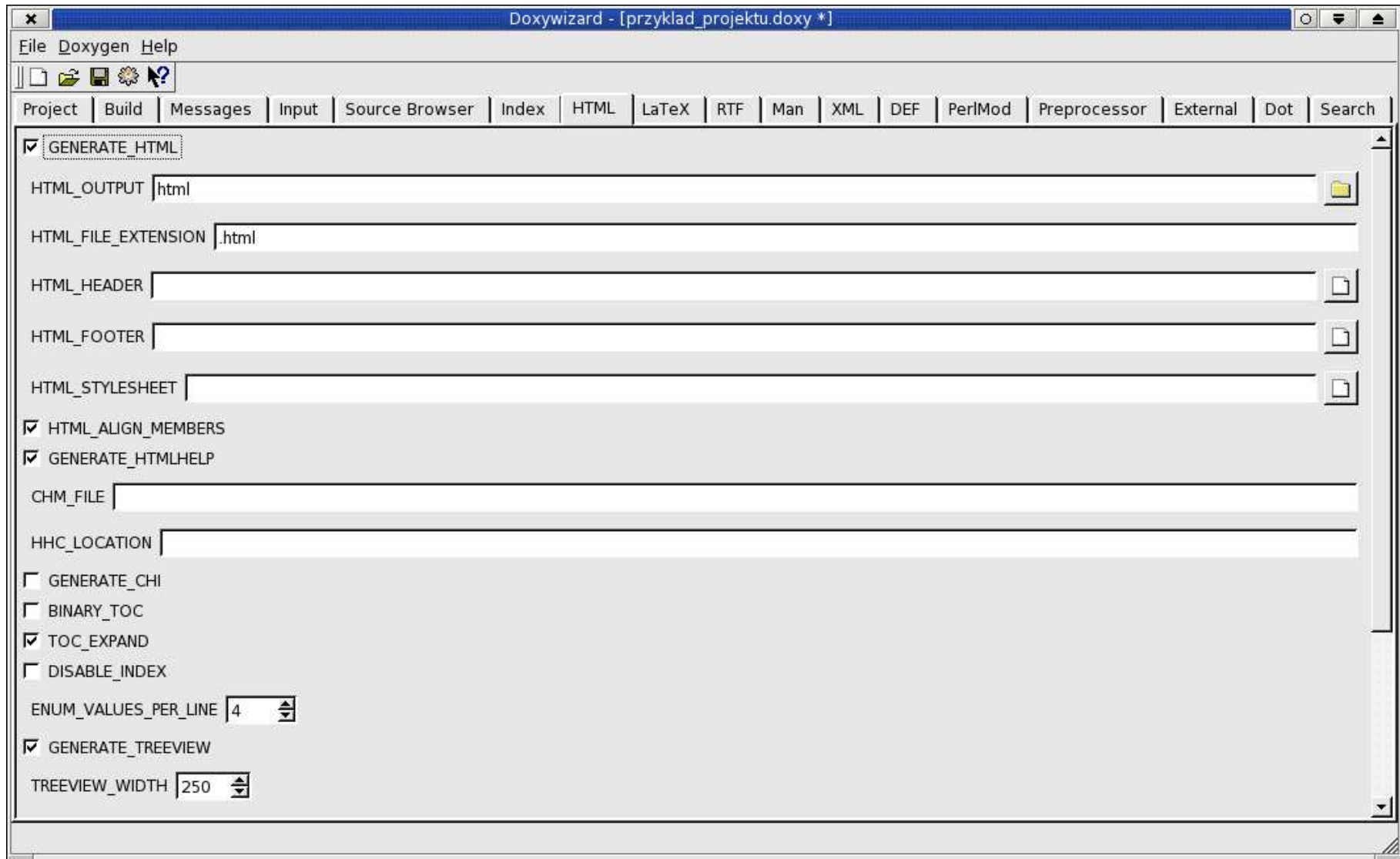
Rysunek 6: Opcje cytowania źródeł.

Doxywizard - tworzenie pliku konfiguracyjnego (cz. 6)



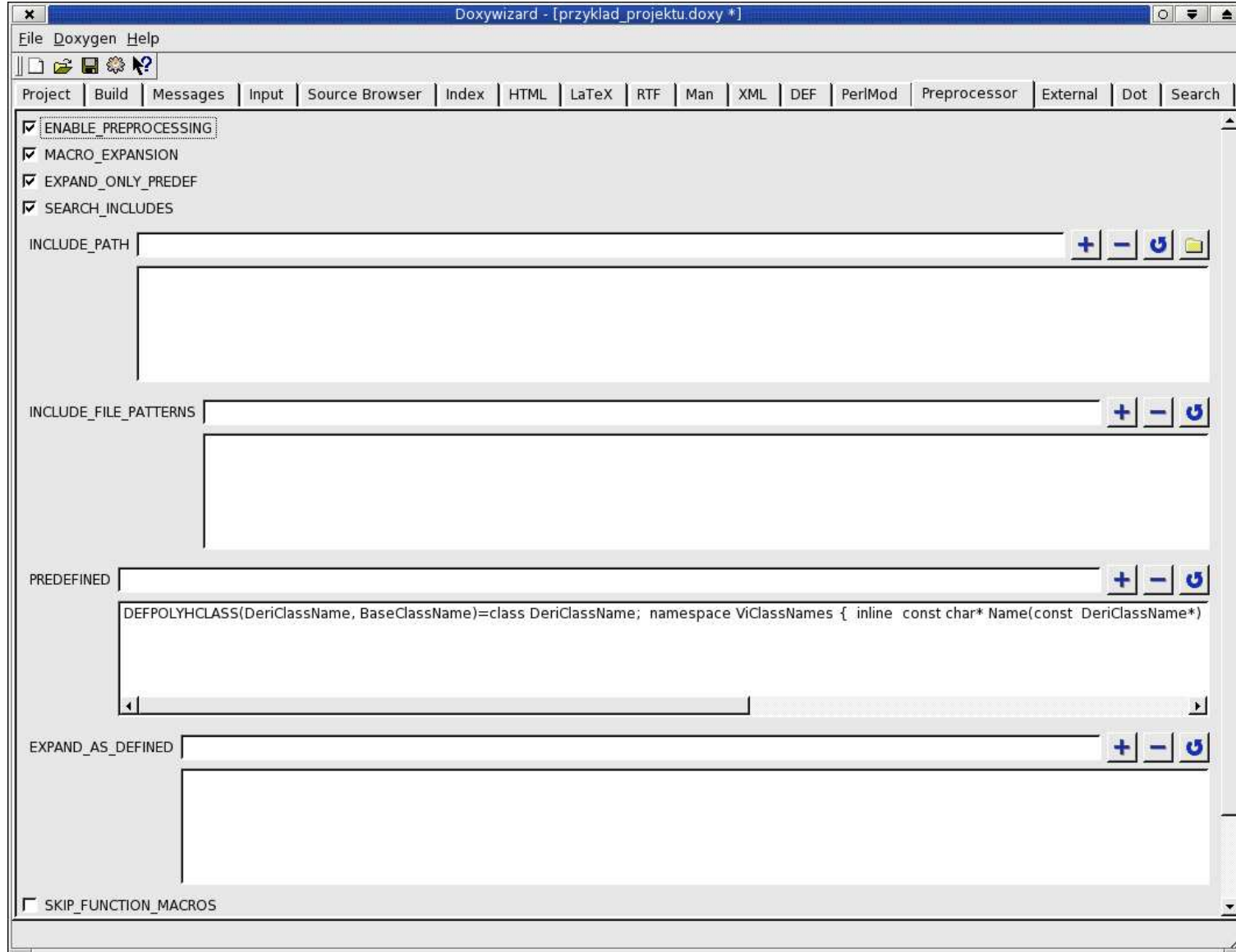
Rysunek 7: Opcje tworzenia indeksu.

Doxywizard - tworzenie pliku konfiguracyjnego (cz. 7)



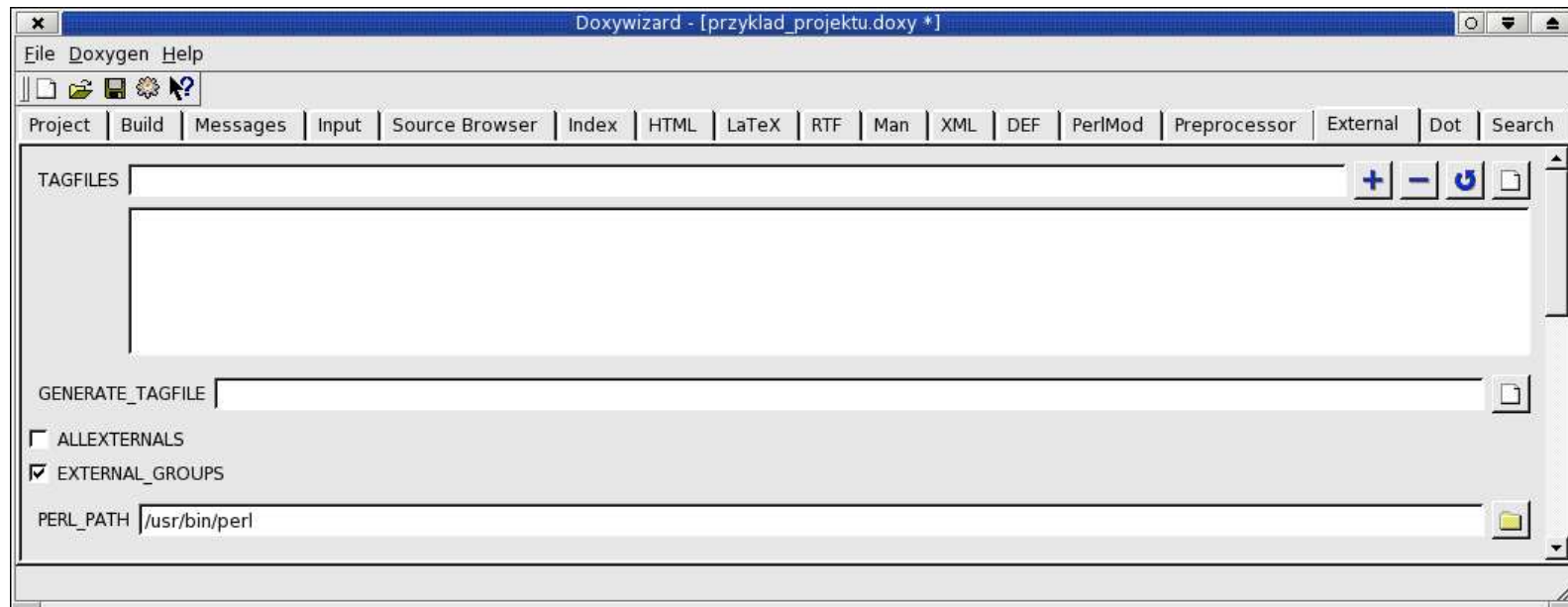
Rysunek 8: Opcje dla dokumentu HTML.

Doxywizard - tworzenie pliku konfiguracyjnego (cz. 8)



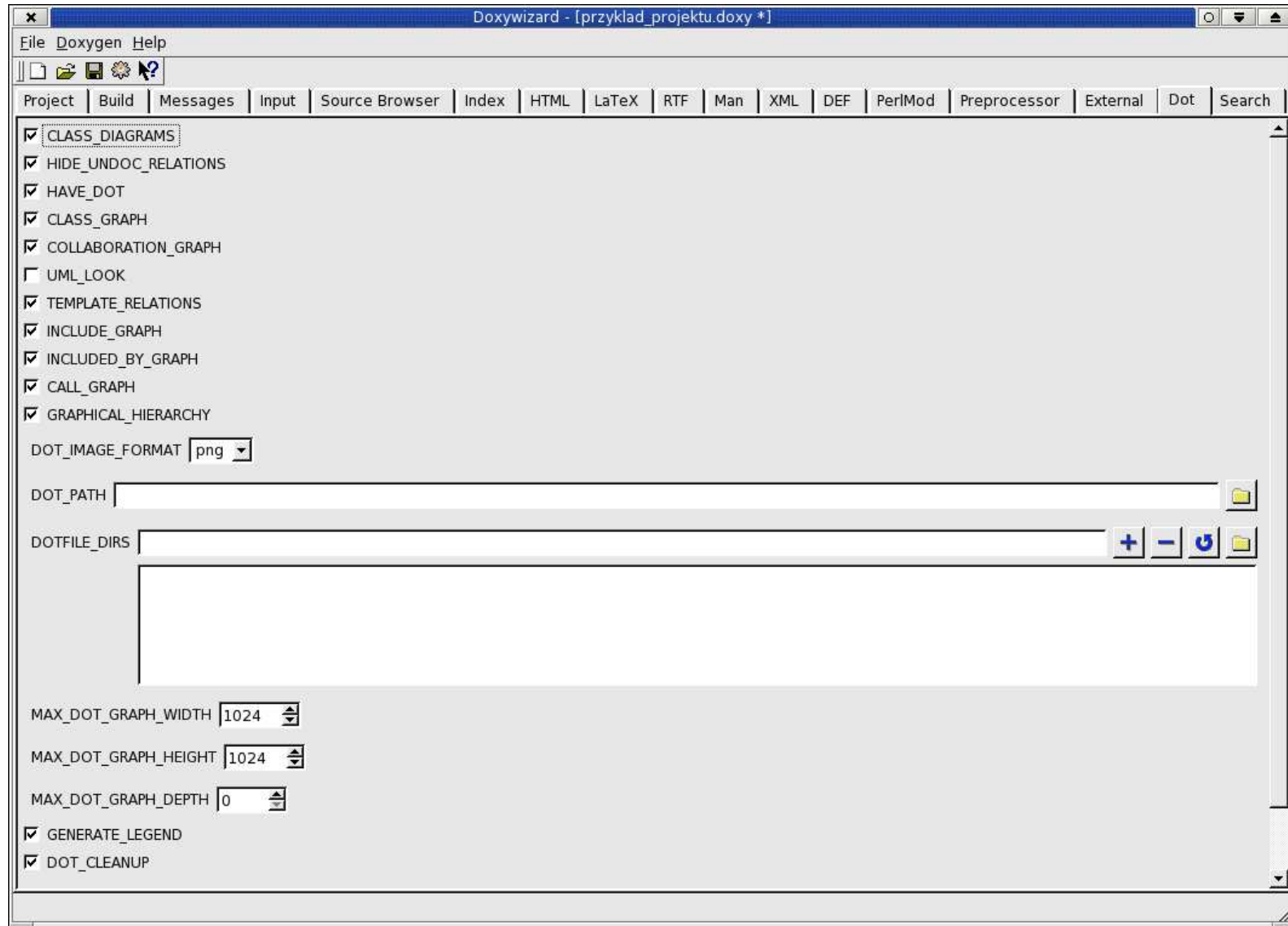
Rysunek 9: Opcje przetwarzania poleceń preprocesora.

Doxywizard - tworzenie pliku konfiguracyjnego (cz. 9)



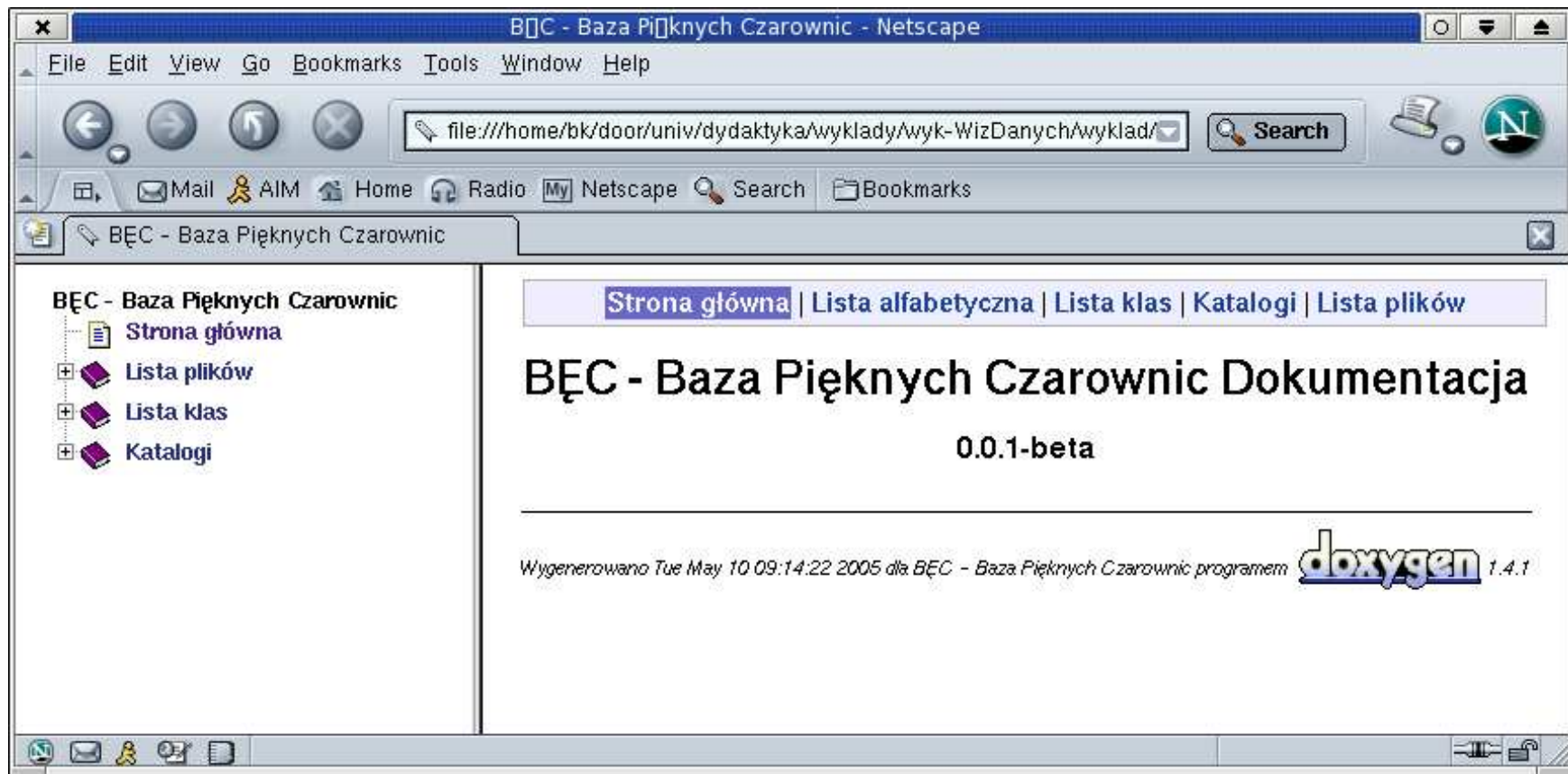
Rysunek 10: Opcje umożliwiające generację dosyłaczy do zewnętrznej dokumentacji.

Doxywizard - tworzenie pliku konfiguracyjnego (cz. 10)



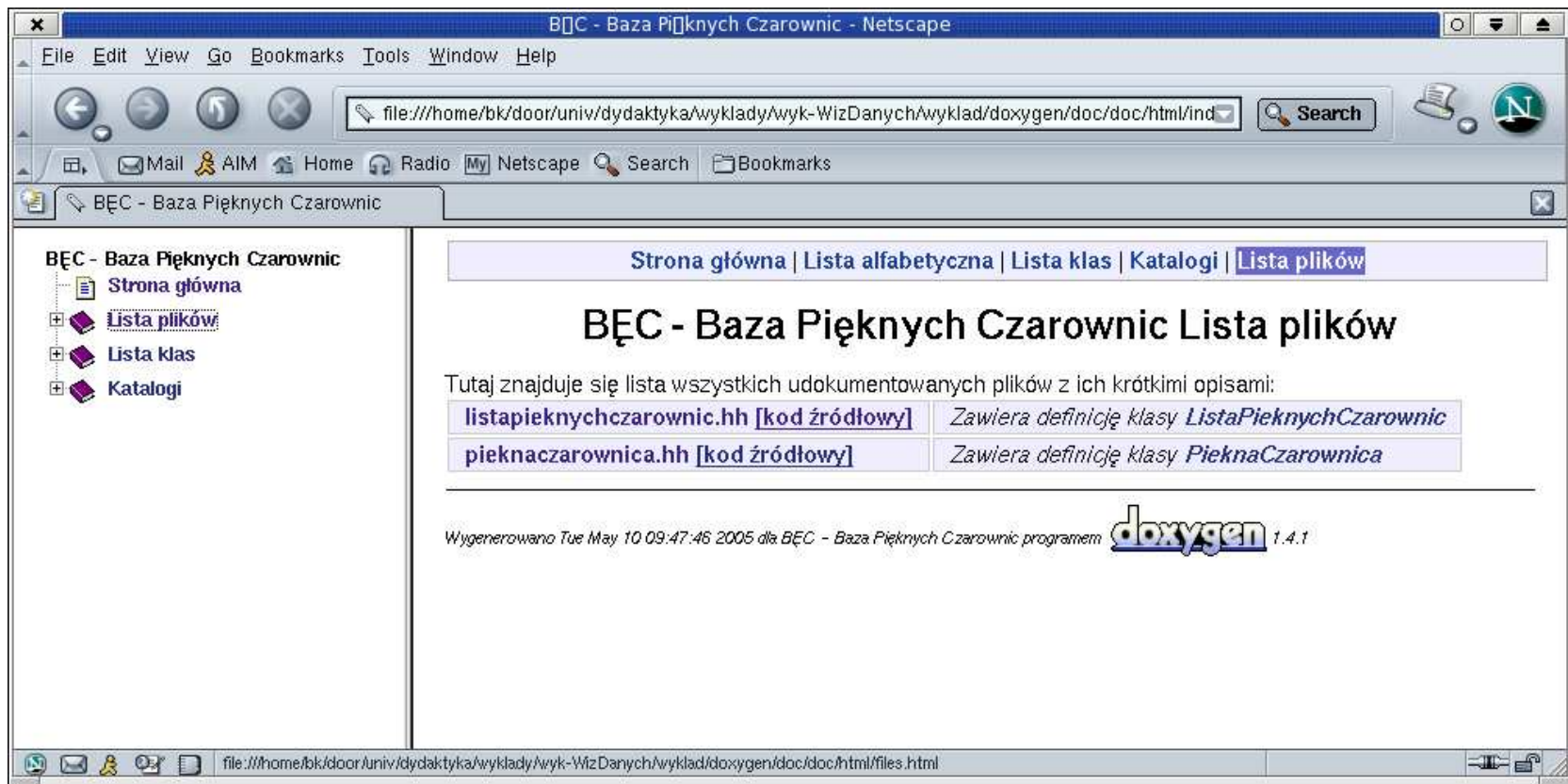
Rysunek 11: Opcje dla generacji grafów.

Doxygen - realizacja dokumentacji



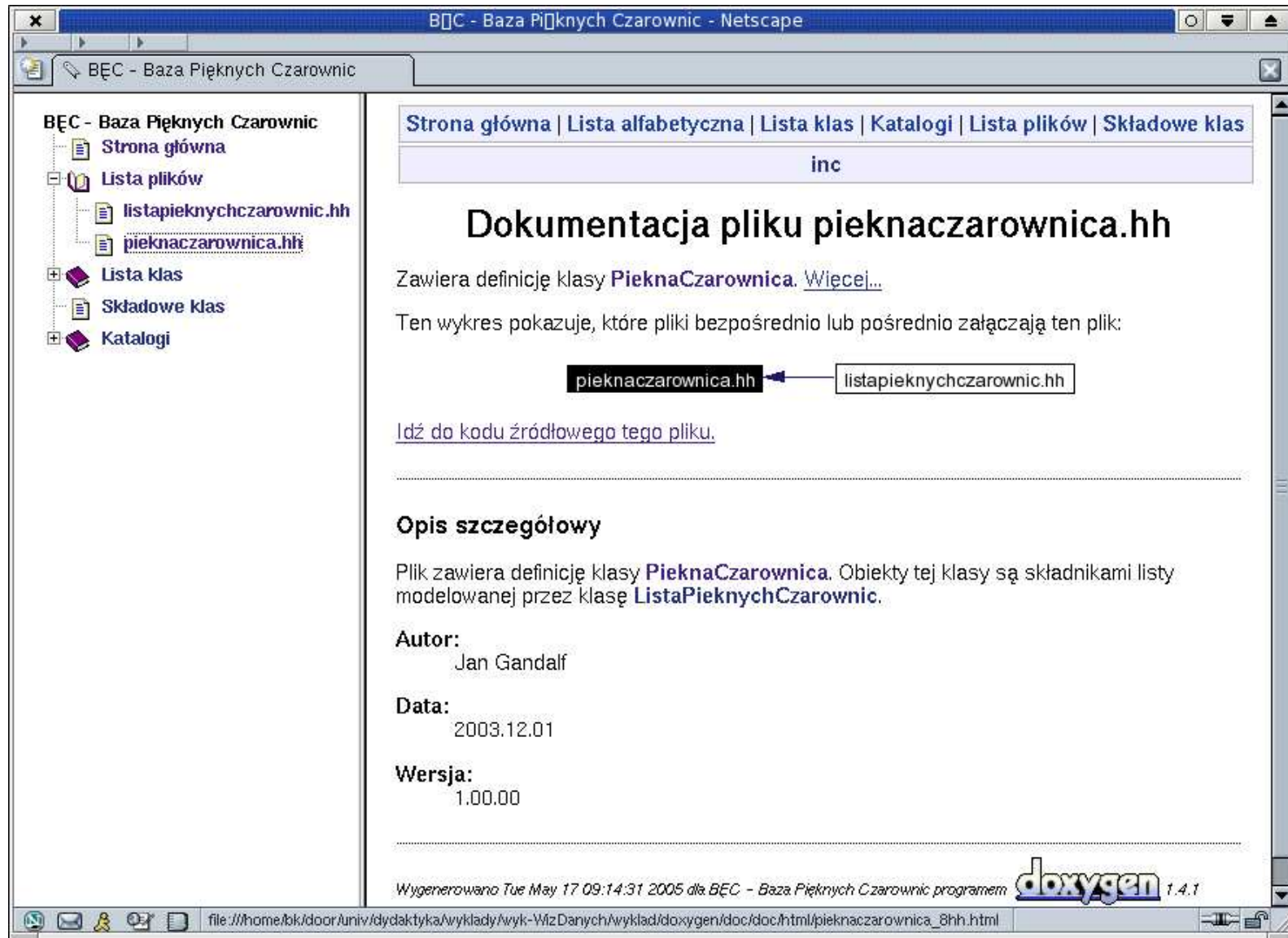
Rysunek 12: Strona tytułowa zdokumentowanego systemu.

Doxygen - realizacja dokumentacji



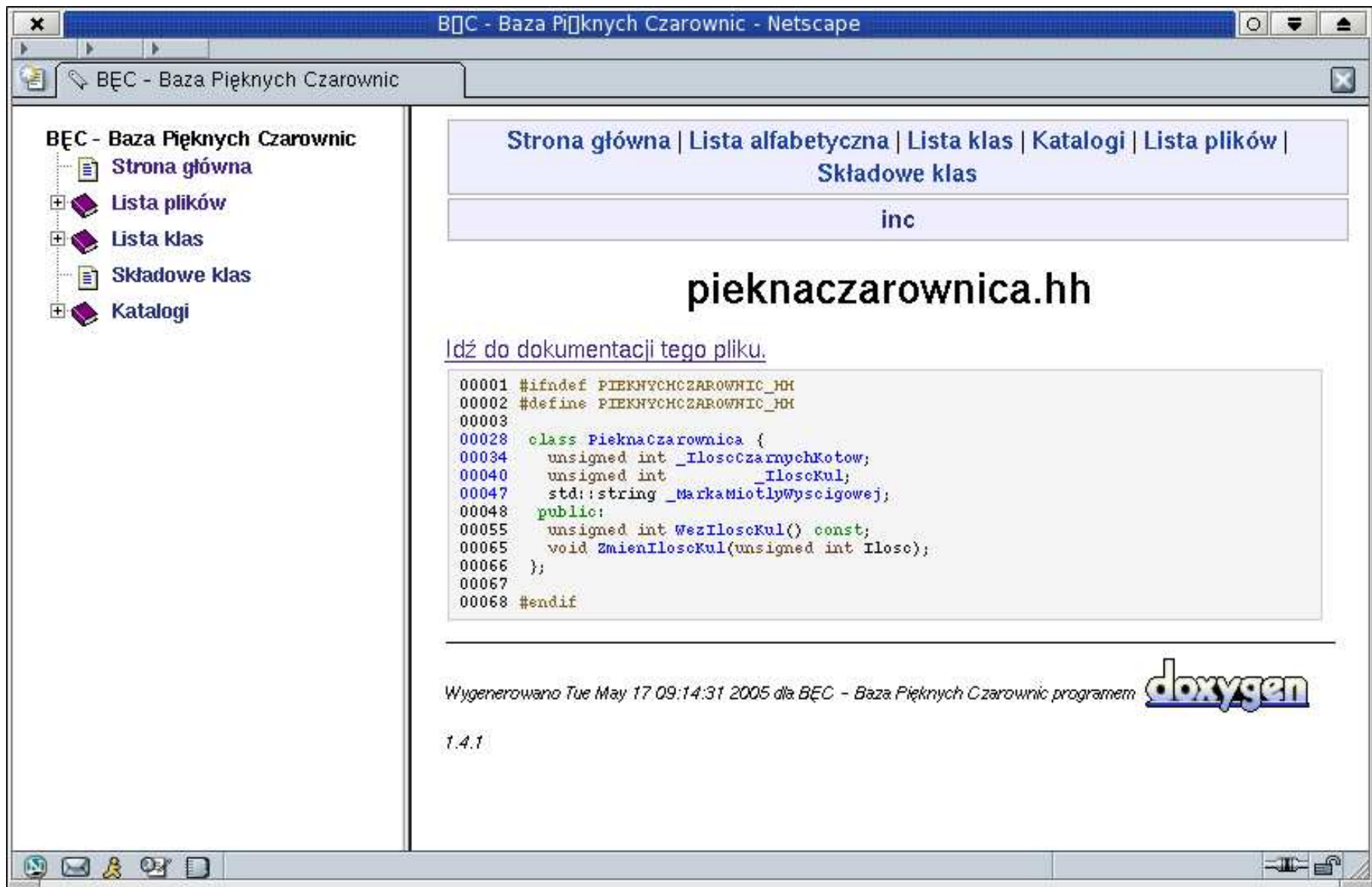
Rysunek 13: Strona dokumentacji plików.

Doxygen - dokumentacja pliku (cz. 1)



Rysunek 14: Strona dokumentacji pliku.

Doxygen - dokumentacja pliku (cz. 2)



Rysunek 15: Strona dokumentacji pliku.

Doxygen - dokumentacja pliku

```
#ifndef PIEKNYCHCZAROWNIC_HH
#define PIEKNYCHCZAROWNIC_HH

/*! \file pieknaczarownica.hh
 *
 * \brief Zawiera definicję klasy PieknaCzarownica
 *
 * Plik zawiera definicję klasy PieknaCzarownica.
 * Obiekty tej klasy są składnikami listy modelowanej przez
 * klasę ListaPieknychCzarownic.
 *
 * \author Jan Gandalf
 * \date 2003.12.01
 * \version 1.00.00
 */

. . .

#endif
```

Doxygen - dokumentacja klasy



Rysunek 16: Strona dokumentacji klasy.

Doxygen - dokumentacja klasy

```
. . .
/ *! \class PieknaCzarownica
* \brief Modeluje piękną czarownicę
*
* Klasa modeluje prawdziwą piękną czarownicę. Atrybutami klasy są
* dane dotyczące ekwipunku pięknej czarownicy tj. szklana kula,
* czarne koty, miotła wyścigowa (patrz opis poszczególnych pól, np.
* \link PieknaCzarownica::_IloscCzarnychKotow
* \endlink)
* Kryterium przynależności do klasy jest subiektywne. Ponadto
* przyjmuje się, że każda czarodziejka jest z definicji piękną
* czarownicą.\n
*/
class PieknaCzarownica {
    . . .
};
. . .
```

Doxygen - dokumentacja pól klasy



Rysunek 17: Strona dokumentacji pól klasy.

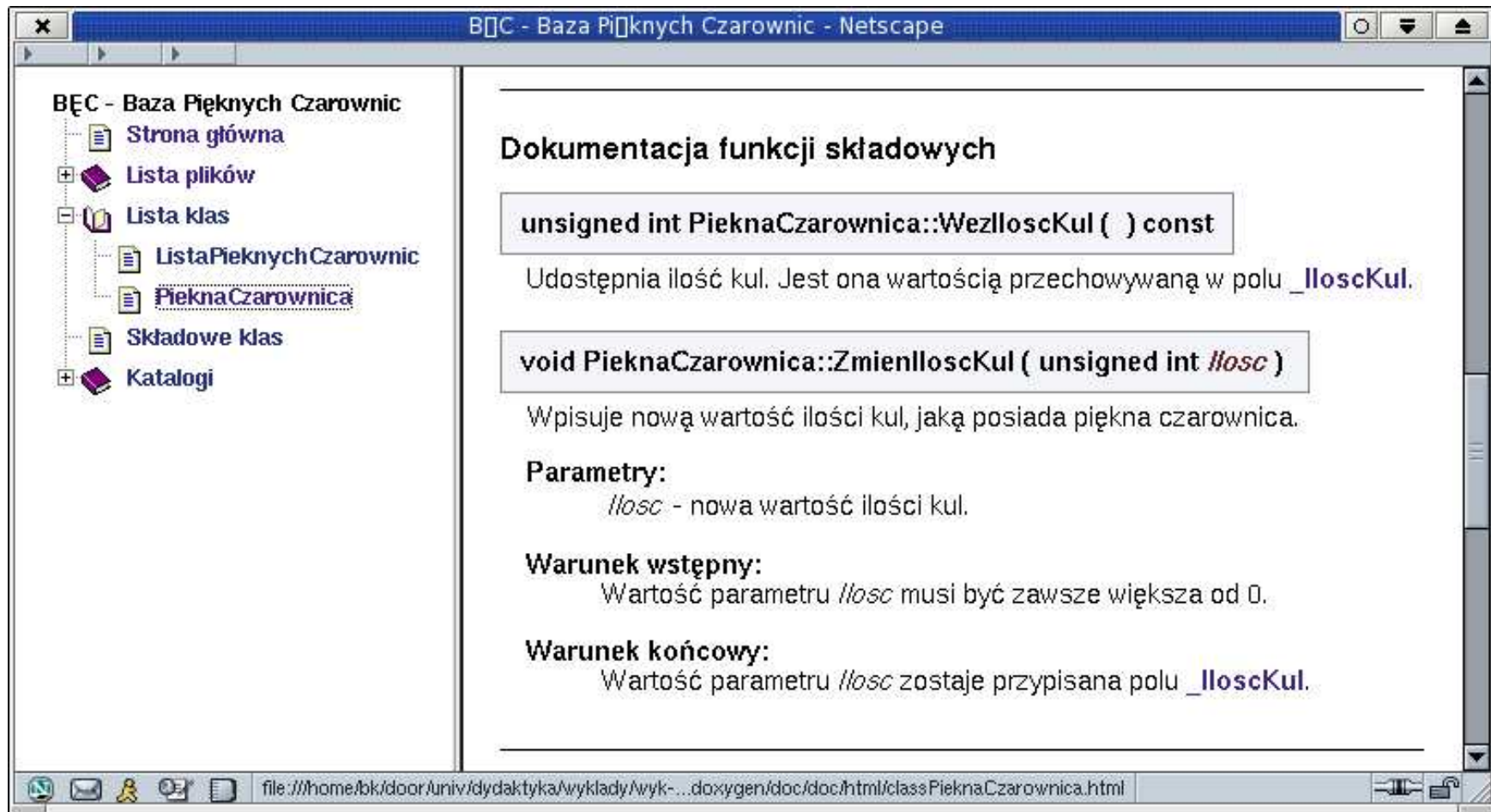
Doxygen - dokumentacja pól

. . .

```
class PieknaCzarownica {  
    /*! \brief Ilość czarnych kotów  
    *  
    * Pole zawiera liczbę czarnych kotów jakie posiada piękna czarownica.  
    * Dopuszcza się, że ilość może być równa 0.  
    */  
    unsigned int _IloscCzarnychKotow;  
    /*! \brief Ilość szklanych kul  
    *  
    * Pole zawiera liczbę szklanych kul. Zakłada się, że ilość musi być  
    * zawsze większa od 0.  
    */  
    unsigned int _IloscKul;  
    /*! \brief Marka i oznaczenie miotły wyścigowej.  
    *  
    * Pole zawiera markę i oznaczenie miotły wyścigowej.  
    * Nazwa pusta oznacza brak miotły.  
    */  
    std::string _MarkaMiotlyWyscigowej;  
    public:  
    . . .  
};
```

. . .

Doxygen - dokumentacja metod



Rysunek 18: Strona dokumentacji metod klasy.

Doxygen - dokumentacja metod

. . .

```
class PieknaCzarownica {
```

```
    . . .
```

```
    public:
```

```
    /*! \brief Udostępnia ilość kul
```

```
    *
```

```
    * Udostępnia ilość kul. Jest ona wartością przechowywaną w  
    * polu \link PieknaCzarownica::_IloscKul _IloscKul\endlink.
```

```
    */
```

```
    unsigned int WezIloscKul() const ;
```

```
    /*!
```

```
    * \brief Zmienia ilość kul
```

```
    *
```

```
    * Wpisuje nową wartość ilości kul, jaką posiada piękna czarownica.
```

```
    * \param Ilosc - nowa wartość ilości kul.
```

```
    * \pre Wartość parametru \e Ilosc musi być zawsze większa od 0.
```

```
    * \post Wartość parametru \e Ilosc zostaje przypisana polu
```

```
    * \link PieknaCzarownica::_IloscKul _IloscKul\endlink.
```

```
    */
```

```
    void ZmienIloscKul( unsigned int Ilosc );
```

```
};
```

. . .