

Doxygen-Howto

1. Einleitung

Um als Neueinsteiger in ein komplexes Projekt einsteigen zu können, ist eine Dokumentation beziehungsweise ein Design-Dokument unabdingbar. Vor allem bei OO-Projekten ist das zugrunde gelegte Datenmodell nur schwer aus dem Source-Code zu extrahieren. Die Klassen-Hierarchie und das zu grunde gelegte Design lässt sich nur mit viel Mühe aus dem Spagetti-Code herauslesen. Ausserdem steht man vor dem Problem, dass neue Entwickler auch neue Erweiterungen beziehungsweise Umorganisationen in der Klassenhierarchie vornehmen, diese im Design-Dokument aber vergessen werden, zu dokumentieren. Um diese Probleme zu umschiffen, wäre eine automatisch erstellte Dokumentation praktisch. An diesem Punkt setzt Doxygen an.

2. Doxygen

Doxygen ist ein Tool, mit dem man Dokumentationen für C++, C, Java, IDL C# automatisch erstellen kann. Dabei steht Doxygen unter der GPL (mehr zur Lizenz unter [1]), was dem Anwender die Möglichkeit anbietet, direkt den Quellcode bearbeiten zu können. Doxygen wird unter Linux entwickelt, man kann es auch unter Windows-Systeme einsetzen. Dieses Howto beschäftigt sich zunächst einmal mit der Linux-Variante.

3. Vorbereitungen fuer den Einsatz

Im Quellcode hinterlassen Entwickler hoffentlich Kommentare, die den Source lesbarer machen sollen. Was läge näher als diese Kommentare so aufzubereiten beziehungsweise anzulegen, dass man aus dieser eine Dokumentation anzulegen. Genau das ist das Prinzip von Doxygen. Die Kommentare im Source werden nach einer gewissen Konvention bearbeitet und später von Doxygen ausgelesen. Aus diesen erstellt Doxygen dann seine Dokumentation. Das Format der Dokumentation wird mittels eines config-Files erzeugt, welches mittels

```
> doxygen -g <config-file>
```

erzeugt wird. Nun wird ein Konfigurationsfile erzeugt, welches einen Haufen an Konfigurationsflags setzt, die momentan allerdings zu grössten Teil uninteressant sind. Wichtig sind vor allem die folgenden Tags:

```
PROJECT_NAME      = <Name_des_Projektes>
OUTPUT_DIRECTORY = <Pfad_fuer_generierte_Doku>
OUTPUT_LANGUAGE   = <Sprache>
INPUT              = <Pfade_zu_den_Sourcen>
FILE_PATTERNS     = <Endung> zum Beispiel .h,.cpp bei einem c++ Projekt
```

Hat man die notwendigen tags im Configure-File gesetzt, kann man die Dokumentation mittels des Kommandos

```
> doxygen <Config-File>
```

erzeugen. Unter Windows nimmt man dass das entsprechende GUI-frontend.

Einige weitere Tags:

```
\struct    ... Dokumentation zu einem C-Strukt
\union     ... Dokumentation zu einer Union
\enum     ... Dokumentation zu einem Enumeration Typ
\fn       ... Dokumentation einer Funktion
\var      ... Dokumentation einer Variable
\def      ... Dokumentation eines #defines
\file     ... Dokumentation eines Files
\namespace ... Dokumentation eines Namespaces
```

5. Abschliessende Bemerkungen

Wer Fragen oder Anmerkungen zu diesem Mini-Howto hat, kann diese direkt an sir_kimmi@gmx.de senden.

Dazu gibt es noch die Anlaufpunkte [2], [3] und [4]. Dazu wird dieses Dokument weiterhin von mir gepflegt.

6. Weiterführende Links

- [1] <http://www.gnu.org>
- [2] <http://www.zfx.info>
- [3] <http://www.zfxengine.de>
- [4] <http://www.sir-kimmi.de>