

Doxygen: Dokumentationswerkzeug für Quelltexte

# Autogene Handbuchmaschine

Jürgen Schuck

Tue Gutes und schreibe darüber – leicht gesagt, wenn man die gesamte schöpferische Kraft für die Entwicklung einzigartiger Software verbraucht hat, und deren Dokumentation in einen Todo-Listeneintrag mündet, dessen sich niemand annehmen will. Hier hilft das frei verfügbare Doxygen, das ansprechende Dokumentation von Software-Projekten unter Unix und Windows quasi ohne zusätzlichen Aufwand erstellt.



Bei Doxygen ([www.doxygen.org](http://www.doxygen.org)) – der Name ist eine Verschmelzung der Begriffe *Documentati*on und *Generat*or – handelt es sich um ein automatisches Analysewerkzeug für Programmstrukturen. Es untersucht Sourcen in C/C++, Java, IDL, C#, Objective C, D sowie PHP und erstellt daraus grafische Darstellungen von Datei- und Klassenabhängigkeiten, verlinkt sämtliche Bezeichner untereinander und formatiert das Ergebnis als HTML-Seiten, Unix Manual Pages (man), Latex und Rich Text Format (RTF). Ebenfalls automatisch generierte Makefiles können die Latex-Dateien nach Postscript und in das Portable Document Format (PDF) überführen.

Zur Steuerung benötigt Doxygen eine Konfigurationsdatei, vorzugsweise im Top-Level-Verzeichnis der Quellen angelegt. Das Format dieser Datei ähnelt einem einfachen Makefile: Sie enthält eine Reihe von Zuweisungen (Tags) der Form `TAGNAME = VALUE [VALUE...]`. Die Bedeutungen der Tags und ihrer Wertsyntax beschreibt das Manual (siehe „Online-Ressourcen“ und „Häufig verwendete Tags“). Das Kommando `doxygen -g <file>` erzeugt eine Konfigurationsdatei, in der sämtliche der zirka 150 Tags enthalten und mit sinnvollen Werten versehen sind. Ohne den Parameter

`<file>` ist das Ergebnis eine Datei namens *Doxygen* (Abbildung 1).

Um einen ersten Eindruck von der Qualität der erzeugten Dokumentation zu erhalten, sollten `EXTRACT_ALL` und `SOURCE_BROWSER` den Wert `YES` bekommen. Wenn die Sourcen in einer Verzeichnishierarchie organisiert sind, muss außerdem `RECURSIVE` auf `YES` stehen, damit der gesamte Source-Baum Berücksichtigung findet.

## Programmstruktur ist überschaubar

Falls `OUTPUT_DIRECTORY` unverändert blieb, enthält das aktuelle Verzeichnis nach dem Aufruf des Kommandos `doxygen` die Directories `html` und `latex` mit den Dokumentationen im jeweiligen Format. Der Start der HTML-Variante erfolgt über die Datei `html/index.html` in einem Browser. An dieser Stelle liegt bereits eine Hyper-

textversion der Programmstruktur mit Bezug auf die Quelltexte vor, die überdies in nahezu allen europäischen Sprachen lesbar ist, wenn der Tag `OUTPUT_LANGUAGE` entsprechend eingestellt ist. Wer will oder muss, kann die HTML-Dokumentation in Microsofts Windows 98 Compressed HTML Help Format (CHM) transformieren. Den

### XTRACT

- Doxygen hilft, mit geringem Aufwand aus C/C++, Java, IDL, PHP- und C#-Quelltexten Dokumentationen in HTML-, Latex-, Postscript-, PDF- und Microsoft-Formaten zu generieren.
- Das Analysewerkzeug extrahiert die nötigen Informationen, unter anderem aus den Klassenabhängigkeiten und den Kommentaren der Quelltexte.
- Gut eignet sich Doxygen auch als Analysewerkzeug für fremde und komplexe Quelltexte.

## ONLINE-RESSOURCEN

Website von Doxygen	<a href="http://www.doxygen.org">www.doxygen.org</a>
Doxygen-Dokumentation und Download	<a href="http://www.stack.nl/~dimitri/doxygen/download.html">www.stack.nl/~dimitri/doxygen/download.html</a>
Dimitri van Heesch's Doxygen Manual	<a href="ftp://stack.nl/pub/users/dimitri/doxygen_manual-1.3.7.pdf.zip">ftp.stack.nl/pub/users/dimitri/doxygen_manual-1.3.7.pdf.zip</a>
Latex-Distribution TeX 1.0	<a href="http://www.tug.org/interest.html#free">www.tug.org/interest.html#free</a>
Ghostscript-Interpreter	<a href="http://www.ghostscript.com/doc/AFPL/index.htm">www.ghostscript.com/doc/AFPL/index.htm</a>
Website des Graph Visualization Project	<a href="http://www.research.att.com/sw/tools/graphviz/">www.research.att.com/sw/tools/graphviz/</a>
Cygwin-Installation	<a href="http://www.cygwin.com/setup.exe">www.cygwin.com/setup.exe</a>
Microsofts HTML Help Workshop Download	<a href="http://msdn.microsoft.com/library/default.asp?url=/library/en-us/htmlhelp/html/vsconHH1Start.asp">msdn.microsoft.com/library/default.asp?url=/library/en-us/htmlhelp/html/vsconHH1Start.asp</a>
Trolltechs Qt-Bibliothek	<a href="http://www.trolltech.com">www.trolltech.com</a>

dazu erforderlichen HTML Help Workshop gibt es umsonst im Web (siehe „Online-Ressourcen“).

Die Tags `HAVE_DOT` und `CALL_GRAPH`, jeweils auf `YES` gesetzt, aktivieren die grafischen Funktionen. Sie ergänzen die Dokumentation um anklickbare Darstellungen von Include-Abhängigkeiten, Klassenhierarchien und -vererbungen sowie Objektrelationen als Interaktionsdiagramme (Collaboration-Diagrams). Vorsichtig einsetzen sollte man `CALL_GRAPH`, da der für jede Funktion und Methode erstellte Graph auch die indirekt aufgerufenen Funktionen enthält, wodurch die Generierung der Dokumentation eine Weile dauern kann. Insbesondere für die Analyse fremder Software kann sich das Warten jedoch lohnen. Jede Änderung der Konfigurationsdatei erfordert allerdings einen erneuten Durchlauf von Doxygen, damit sie sich in der Dokumentation niederschlägt.

Interaktive Grafen erzeugt Doxygen mit dem `dot`-Werkzeug aus dem Graph Visualization Project Graphviz der AT & T und Lucent Bell Labs, das dazu installiert sein muss (siehe „Hinweise zur Installation“). Dabei repräsentieren Kombinationen von Farben

und Linienarten die verschiedenen Arten von Abhängigkeiten wie *public*, *private*, *ruft auf* oder *wird aufgerufen*. Der Tag `GENERATE_LEGEND` fügt jeder Grafik einen Link zu einer ausführlichen Legende hinzu, was den Umgang mit der Dokumentation weiter vereinfacht. Voraussetzung hierfür ist, dass `GENERATE_LEGEND` auf `YES` steht.

## Nach Bedarf: Latex oder Postscript

Das Verzeichnis `latex` enthält die Latex-Quellen der Dokumentation, für deren Verarbeitung eine Latex-Distribution notwendig ist. Neben den Sourcen liegt hier ein `Makefile`, mit dem ein `make`-Aufruf die Datei `refman.dvi` erzeugt. Je nach Bedarf erstellen `make ps` und `make pdf` Postscript- beziehungsweise PDF-Versionen der Dokumentation. Für die Postscript-Variante sollte der Tag `PDF_HYPERLINKS` auf `NO` stehen. Der Wert `YES` schaltet die Generierung von Hyperlinks im PDF-Dokument ein. Eine Ausgabe im PDF-Format setzt den Postscript-Interpreter Ghostscript voraus, da `make pdf` das enthaltene `ps2pdf` verwendet.

Mit `GENERATE_MAN = YES` schaltet der hartgesottene Unixler die Er-

## Hinweise zur Installation

Wie aufwendig sich die Installation von Doxygen gestaltet, hängt vom verwendeten Betriebssystem ab. Für Linux (i386) und Mac OS X (ab Panther-Release) gibt es im Download-Bereich von [www.doxygen.org](http://www.doxygen.org) fertige Binaries als komprimierte Tar-Archive. Für Windows ist zusätzlich ein Installer verfügbar.

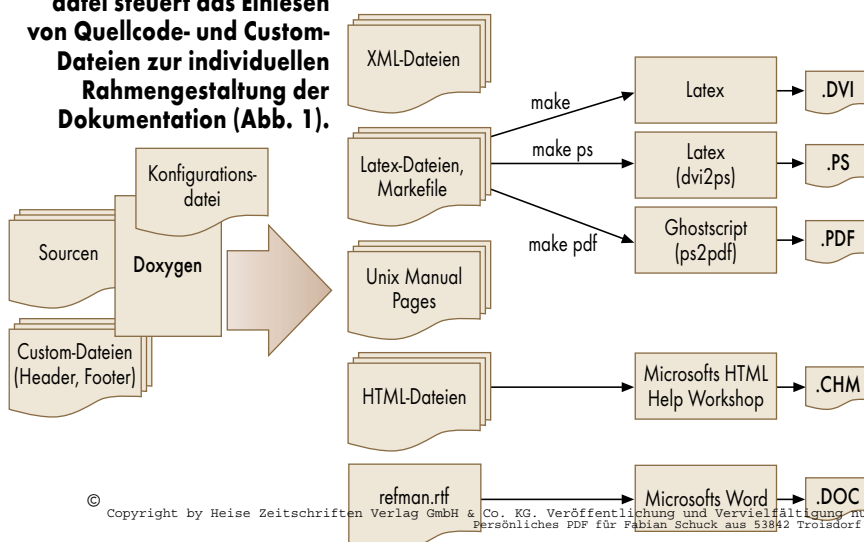
Wer unter einem anderen Unix-Derivat arbeitet, muss die Quellen übersetzen, was aufgrund des `configure`-Skripts einfach sein dürfte. Die Entwicklung von Doxygen erfolgt unter Linux, und die Übersetzung setzt einige Gnu-Werkzeuge in bestimmten Versionen voraus: GCC 2.95.3, Flex 2.5.4, Bison 1.35 sowie Perl 5.005\_03. Eine Übersetzung der Sourcen unter Windows mit Cygwin ist nicht möglich, Voraussetzung ist Visual C++ 6.0.

In einigen Latex-Distributionen ist der Papiertyp A4 (`a4wide.sty`) nicht enthalten. In diesem Fall kann in der Konfigurationsdatei das Tag `PAPER_TYPE` auf einen unterstützten Wert gesetzt werden.

Das Graph Visualization Project muss ab Doxygen 1.3 mindestens in der Version 1.8.10 installiert sein. Die Software ist für eine Reihe von Unixen, darunter Mac OS X, Linux und FreeBSD, in Binärversionen verfügbar.

Unter Windows empfiehlt sich für einen schnellen Start die Installation von Cygwin mit den `texex`-, `ghostscript`- und `doxygen`-Packages. Für das Graph Visualization Project gibt es bei Lucent einen Installer für Windows. Nach der Installation muss sichergestellt sein, dass die Datei `dot.exe` sich im Pfad der Shell befindet. Die aktuelle Doxygen-Version ist der gegenwärtigen Cygwin-Version um ein Minor-Release voraus und sollte daher bevorzugt installiert werden.

### Doxygens Konfigurationsdatei steuert das Einlesen von Quellcode- und Custom-Dateien zur individuellen Rahmengestaltung der Dokumentation (Abb. 1).



zeugung von Manual Pages ein und findet diese nach dem Doxygen-Lauf im Verzeichnis `man`. Zwar enthalten die generierten `man`-Pages keine Grafiken und Hyperlinks, sehen aber ansonsten genauso aus, wie man sie kennt. Zum Ein- und Ausschalten der unterstützten Ausgabeformate dienen neben `GENERATE_MAN` die Tags `GENERATE_HTML`, `GENERATE_HTMLHELP`, `GENERATE_LATEX` und `GENERATE_RTF`.

Mit den bis hierhin beschriebenen Möglichkeiten lassen sich bereits griffige Dokumentationen erstellen, die zum Verständnis insbesondere von fremden Software-Projekten beitragen können. Deren Eingängigkeit kann die

## HÄUFIG VERWENDETE TAGS

INPUT	Liste der Source-Dateien und Verzeichnisse. Voreingestellt sind *.c, *.cc, *.cxx, *.cpp, *.c++, *.java, *.ii, *.ixx, *.ipp, *.i++, *.inl, *.h, *.hh, *.hxx, *.hpp, *.h++, *.idl, *.odl, *.cs, *.php, *.php3, *.inc, *.m und *.mm
EXCLUDE	Ausschluss von Dateien und Verzeichnissen
RECURSIVE	Source-Verzeichnishierarchie rekursiv durchlaufen
EXTRACT_ALL	Auch nicht dokumentierte Dateien (im Sinne von Doxygen) berücksichtigen
SOURCE_BROWSER	Dokumentation mit Hyperlinks zu den Quelltexten generieren
OUTPUT_DIRECTORY	Top-Level-Verzeichnis der generierten Dokumentation(en)
OUTPUT_LANGUAGE	Ausgabesprache. Zum Beispiel German, English, French, Czech, Dutch, Norwegian
GENERATE_HTML	Dokumentation im HTML-Format erzeugen
GENERATE_HTMLHELP	Compressed-HTML-Help-Format (CHM) erzeugen
GENERATE_RTF	Rich Text Format-Format erzeugen
GENERATE_LATEX	Latex-Format erzeugen
GENERATE_MAN	Unix Manual Pages erzeugen
HAVE_DOT	Verfügbarkeit des dot-Werkzeugs aus Graphviz
CALL_GRAPH	Graph der Funktions- und Methodeaufrufe erzeugen (kann bei größeren Projekten eine Weile dauern)

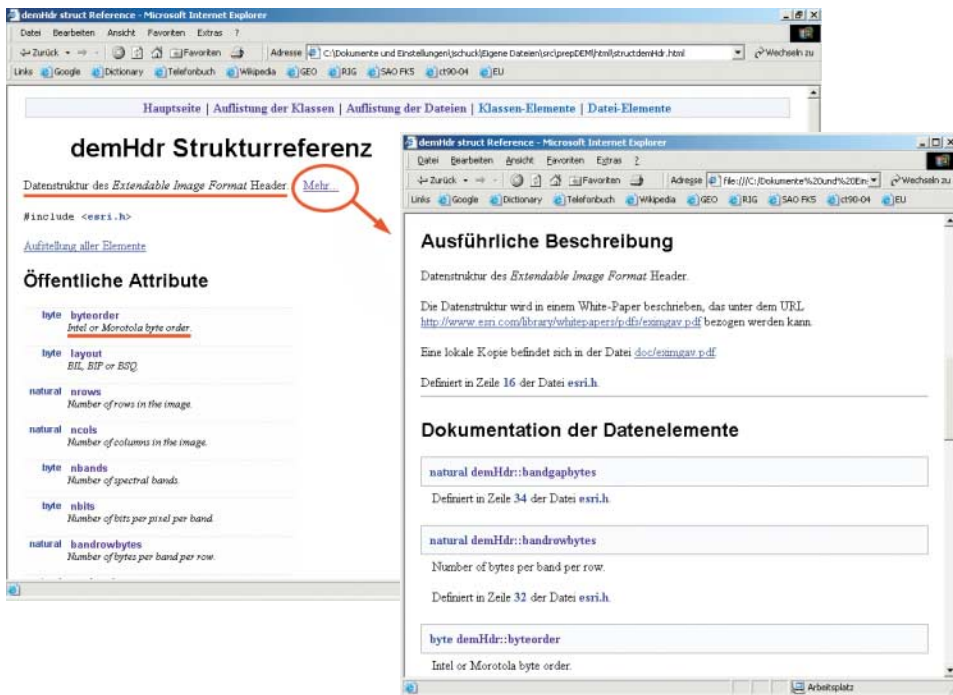
Übernahme von Kommentaren aus den Sourcen noch steigern. Dazu muss der Programmierer sie in der folgenden Form schreiben:

```
/*!
 * ... Kommentar ...
 */
```

Daneben sind weitere erlaubt, die sich durch die Wahl und Anordnung der Kommentarzeichen unterscheiden. Kommentare zu Sprachelementen wie Klassen, Namensräume, Methoden, Funktionen, Variablen, Aufzählungen und Typen der unterstützten Programmiersprachen müssen vor deren Deklarationen oder den Definitionen stehen, damit Doxygen sie erkennt und berücksichtigt. Das Werkzeug strukturiert Kommentare in Überschriften und Texte, die wiederum durch Absätze untergliedert werden können. Überschriften leitet die Zeichenfolge `¶` ein. Diese speziellen Kommandos, von denen es eine ganze Reihe gibt, kann anstelle des Backslash auch ein `@`-Zeichen einleiten. Überschriften enden bei einer leeren Zeile, dem Absatztrenner.

Listing 1 zeigt eine Include-Datei mit der Definition eines Strukturtyps. Die von Doxygen aufgrund ihrer Formate berücksichtigten Kommentare sind hervorgehoben. Der Kommentarblock in der Mitte entspricht keinem unterstützten Format und taucht daher in der erzeugten Dokumentation nicht auf.

Der erste Kommentarblock definiert eine Überschrift (`@brief`), auf die zwei Absätze (Leerzeilen) mit Text folgen.



**Das Dokumentationsergebnis aus Listing 1: Im hinteren Fenster sieht man die Überschrift mit dem kursiven „Extendable Image Format“, das vor-dere zeigt den zugehörigen Text mit den automatisch und manuell eingefügten Hyperlinks (Abb. 2).**

```
Listing 1: esri.h

#ifndef _ESRI_H
#define _ESRI_H

typedef unsigned char    byte ;
typedef unsigned long long int natural ;
typedef long double      real ;

/*!
 * @brief Datenstruktur des @e Extendable @e Image @e Format Header.
 *
 * Die Datenstruktur wird in einem White-Paper beschrieben, das unter dem URL
 * http://www.esri.com/library/whitepapers/pdfs/eximgav.pdf bezogen werden kann.
 *
 * Eine lokale Kopie befindet sich in der Datei <A HREF="file:///c:/dokumente und einstellungen/jeschuck/eigene dateien/src/prepdem/doc/eximgav.pdf">doc/eximgav.pdf</A>.
 */
typedef struct {
#define DH_BYTORD_M 0    ///< Motorola, big-endian.
#define DH_BYTORD_I 1    ///< Intel, little-endian.
    byte byteorder ;    ///< Intel or Motorola byte order.

    /*
     * Details:
     * Detailed format descriptions are located under the URL
     * http://gis.washington.edu/cfr250/lessons/remote_sensing/bil.html.
     */
#define DH_LAYOUT_BIL 0    ///< Band interleaved by line (BIL)
#define DH_LAYOUT_BIP 1    ///< Band interleaved by pixel (BIP)
#define DH_LAYOUT_BSQ 2    ///< Band sequential (BSQ)
    byte layout ;          ///< BIL, BIP or BSQ.
    natural nrows ;        ///< Number of rows in the image.
    natural ncols ;        ///< Number of columns in the image.
    byte nbands ;          ///< Number of spectral bands.
    byte nbits ;           ///< Number of bits per pixel per band.
    natural bandrowbytes ; ///< Number of bytes per band per row.
    ...
} demHdr ;

#endif /* _ESRI_H */
```



Das Kommando `@e` bewirkt die kurssive Darstellung des Extendable Image Format in der Dokumentation. Die URL ersetzt ein Hyperlink zu der angegebenen Adresse. Im zweiten Abschnitt sieht man, wie sich an beliebigen Stellen HTML-Code einfügen lässt.

Aus Gründen der Übersichtlichkeit des Quellcodes stehen die Kommentare der Makros und der Strukturelemente hinter den Bezeichnungen, für Doxygen erkennbar durch das `<-`-Zeichen. Die Wahl des Kommentarformats `///` anstelle von `/*!` spart das schließende `*/`. Abbildung 2 zeigt das Ergebnis.

## Wenig Kommentare in Include-Dateien

Die bisher besprochenen Kommandos, von denen es rund 100 gibt, dienen der Formatierung des erzeugten Textes. Für den Fall, dass der Kommentar zur Dokumentation von beispielsweise einer Klasse oder einer Methode nicht unmittelbar vor oder hinter dem Objekt stehen soll, sondern an anderer Stelle oder sogar in einer anderen Datei, bedarf es einer Möglichkeit zur Benennung des gemeinten Objekts. Dies erfolgt über eine Reihe von Kommandos, die Doxygen Name und Typ des betreffenden Objekts mitteilen. Zwar dürfte meist die Kommentierung beim Objekt erfolgen, es sind aber Fälle vorstellbar, in denen dies nicht erwünscht ist. Beispielsweise könnte man auf die Idee kommen, eine Klassendeklaration in einer Include-Datei zunächst nur mit einer Überschrift zu versehen. Bei der Klassendefinition würde man dann die ausführliche Beschreibung anbringen, was die Include-Datei kompakt hielte. Die Listings 2 und 3 zeigen den Mechanismus anhand einfacher Funktionen.

In Abbildung 3 ist das Ergebnis der beiden Listings zu sehen. Die Dokumentation der Include-Datei im hinteren Fenster zeigt unter den Prototypen die Überschriften. Über die Hyperlinks der Funktionsnamen gelangt man zu deren Beschreibungen, die in der C-Datei erfolgten. Bei dieser Art der Kommentierung ist die Voraussetzung dafür, dass Doxygen den richtigen Bezug herstellen kann, dass das Kommando `@file` die Datei mit den Überschriften – in diesem Fall `gtopo30.h` – kennzeichnet.

Das Beispiel verdeutlicht außerdem die Wirkung der Kommandos `@param` und `@return` zur Dokumentation von Funktionsparametern und -rückgabewerten. Schließlich bleibt noch der

**Listing 3: Ausschnitt aus `gtopo30.c`**

```
...
/*! @fn int openG30(const char *path, int flags, demHdr *demInfo)
 * Die Funktion erwartet das Digitale Höhenmodell im angegebenen Verzeichnis in der Form,
 * die unter dem URL http://edcftp.cr.usgs.gov/pub/data/gtopo30/global/w100n90.tar.gz
 * bezogen werden kann.
 *
 * Beim Öffnen des DEM werden die ersten 9 Kacheln mittels der popen() Funktion geöffnet,
 * um sie mit zeilenweise mittels der Funktion readG30 einlesen zu können.
 *
 * Aufgrund seiner Größe kann das gesamte DEM nicht in den Hauptspeicher geladen werden.
 *
 * @param path Verzeichnis mit GTOPO30 Distribution.
 * @param flags Attribute beim Öffnen.
 * @param demInfo Extendable Image Format Header.
 * @return 0 bei Fehler.
 */
int openG30(const char *path, int flags, demHdr *demInfo)
{
    struct stat s ;
    int i, j, k ;

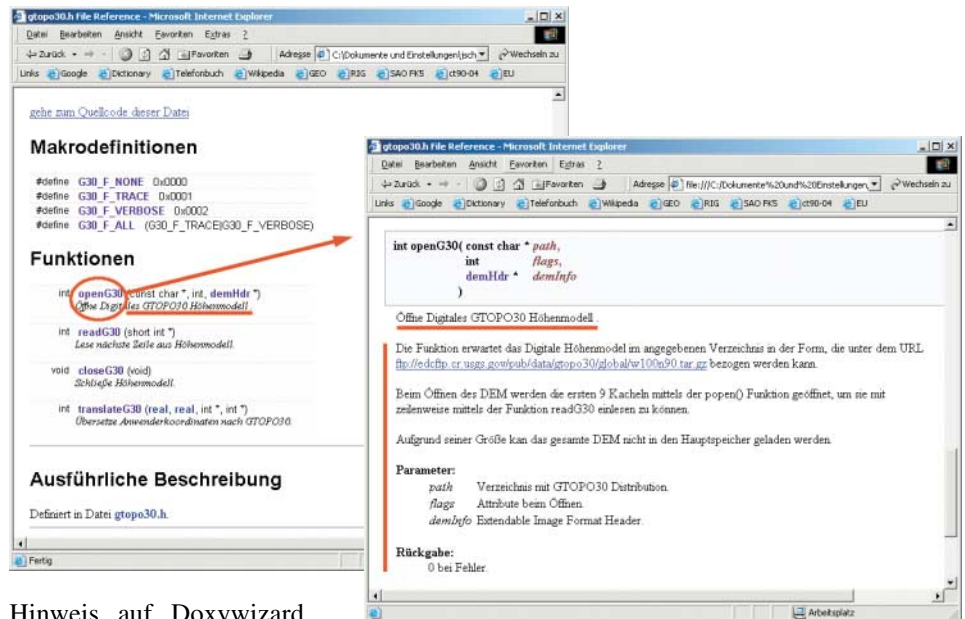
    if ( currentTileRow ) // GTOPO30 ist bereits geöffnet.
        return 0 ;

    ...
}
```

**Listing 2: `gtopo30.h`**

```
/*! @file
 *
 * #ifndef _GTOPO30_H
 * #define _GTOPO30_H
 *
 * #include "esri.h"
 *
 * #define G30_F_NONE 0x0000
 * #define G30_F_TRACE 0x0001
 * #define G30_F_VERBOSE 0x0002
 * #define G30_F_ALL (G30_F_TRACE|G30_F_VERBOSE)
 *
 * /// @brief Öffne Digitales GTOPO30 Höhenmodell .
 * extern int openG30(const char * , int, demHdr *);
 * /// @brief Lese nächste Zeile aus Höhenmodell.
 * extern int readG30(short int *);
 * /// @brief Schließe Höhenmodell.
 * extern void closeG30(void);
 * /// @brief Übersetze Anwenderkoordinaten nach GTOPO30.
 * extern int translateG30(real, real, int *, int *);
 *
 * #endif /* _GTOPO30_H */
```

**Die Header-Datei (Listing 2) enthält die im hinteren Fenster zu sehenden Überschriften, während die C-Datei (Listing 3) die zugehörige Erklärung liefert. Außerdem im hinteren Fenster zu sehen: der Hyperlink zum Quelltext der Datei, da der Tag `SOURCE_BROWSER` in der Konfigurationsdatei den Wert `YES` hat (Abb. 3):**



Hinweis auf Doxywizard, einem GUI-Frontend zur Handhabung der Konfigurationsdateien. Es basiert auf der plattformübergreifenden Qt-Bibliothek von Trolltech, mit der sich einheitliche Bedienoberflächen für Unix und Windows erstellen lassen. Qt gibt es in einer kommerziellen und einer freien Variante. Letztere ist Bestandteil der Binärdistribution von Doxygen für Windows, die Doxywizard enthält. In der Linux-Distribution fehlt das GUI-Frontend, ist jedoch als zusätzliche Komponente über die Doxygen-Website in ausführbarer Form erhältlich.

Doxygen ist ein schnell und einfach zu handhabendes Werkzeug, mit dem Programmierer ohne nennenswerten Mehraufwand griffige Dokumentationen erstellen können. Außerdem hilft die Software dabei, in relativ kurzer Zeit einen Überblick über unbekannte Quelltexte zu gewinnen. (ka)

**JÜRGEN SCHUCK**

ist Mitarbeiter der Materna GmbH und als Projektleiter im Bereich IT-Service-Management tätig.

