

Documenting your PHP code with PHPDoc and DocBook XML

Gábor Hojtsy

International PHP Conference 2002
November 3-6. 2002.

Questions

- Who used/know about PHPDoc?
- JavaDoc?
- DocBook XML?
- XML at all?
- Anyway who documents his/her code regularly using any format?

About myself

- Gábor Hojtsy
- goba@php.net
- Student at Budapest University of Technology and Economics
- PHP.net Webmasters Team
- PHP Documentation Team [build system, XSLT rendering, CHM format, Hungarian translation]

Session outline

- Documentation types
- Developer documentation
- Possible formats
- Inline documentation in PHP code
- Standalone documentation in DocBook
- Tools to handle the two and merge contents

Documentation types

- Software written in PHP
- Plans, concepts, specification [UML]
- As the software is stable
 - User's documentation
 - Help buttons on web pages
 - Support pages, simple manuals
 - Developer's documentation

Developer's documentation

- General information
 - Authors, License, Links
- Concepts explained
 - Reasons to develop this software
 - Technologies, libraries used
- Implementation details
 - Meta information about classes, functions, variables, constants; with usage examples
- Historical information
 - Previously supported API[s], future directions, todo

Expected qualities

- All information in one documentation
- Easily accessible / modifiable
 - Industry wide used format
 - Cross platform tools for viewing and editing
- Preferably with version handling support

Proprietary formats

- Microsoft Word or StarOffice
 - Editors and viewers commonly available
 - Weak version handling, not easily modifiable
- Adobe Portable Document Format
 - More portable than the previous formats
 - Editors are not that widely accessible
- Other proprietary formats
 - Hard to find tools to edit

Write documentation into code

- Add / modify the docs while you write code
- Read documentation as you look at code
- Text format is portable and “moves with code”
- Version handling is supported for text files
- Different output formats can be generated from inline documentation

Documentation comments

- Comments are used to hold doc information
- Comment format is defined so it is parseable with specific tools
- Not all comments are part of the documentation
- Entities are documentable, not every single line of code

Information in documentation

- Metadata about code
 - Authors, license, version, crosslinks
- Explanation of code elements
 - Usage examples, long descriptions
- Restrictions not supported in PHP
 - Variable type information
 - Class member accessibility
- Grouping into packages, subpackages

Major comment formats

- **PHPDoc**
 - Derived from Javadoc
 - Standard format of PEAR
 - Parsers implemented to generate reports
 - IDEs support this “standard”
- **eZ phpdoc**
 - Cryptic format for the first look
- **Docreader**
 - XML formatted comments

eZ phpdoc example

```
//!!! eZCommon
//! The eZTextTool class provides text utility
  functions
/*!
  This class consists of static functions for
  formatting of text. These functions are made as an
  extension to PHP, i.e. functions you would use all
  the time, but aren't part of PHP.
  \code
  $boolean = false;
  echo eZTextTool::boolText($boolean); // prints
  "false"
  \endcode
  \sa eZURITool
*/
class eZTextTool { ...
```

Docreader example

```
/*! <method name="parse" access="public" >  
<summary>Transforms the given XML data into  
an HTML page (NOT complete with header  
and footer), using the saf.XML.Sloppy  
parser. Returns false on  
failure. </summary>  
<param name="xml" type="string" />  
<param name="namespace" type="string"  
default="saf" />  
<returns type="string" />  
</method> ! */  
function parse ($xml , $namespace = 'saf' ) {  
...
```

PHPDoc examples

*Presented live in PHPEdit
and in the Zend Development
Environment*

PHPDoc parsers

- **PEAR/PHPDoc**
 - Initially created by Ulf Wendel
 - Generates it's own XML and HTML output
- **PHPDoc (in Java)**
 - Converts PHP classes and comments to Java classes and comments
 - Feeds the generated files to Javadoc
 - Output is customizable through Doclets

PHPDoc parsers II.

- **phpDocumentor**
 - Written in PHP from scratch
 - Output is customizable through converters
 - HTML, CHM and PDF (alpha) converters bundled
 - Command line and web based version
- **PHPEdit**
 - Generates it's own HTML output set, customizable through XSL style sheets

PHPDoc parsers III.

- **Zend Development Environment**
 - Parses comments for function hints
 - Has it's own phpdoc flavor
- **PHP Code Doc**
 - Browseable color coded PHP source inside the documentation
 - Part of the PHPMole project

Inline documentation problems

- Parsers interpret some parts differently
 - Precise formatting is possible with HTML elements, handled differently by parsers
 - Conceptual information cannot be expressed
- ➔ *We do need a separate documentation for conceptual information*

Conceptual documentation

- Still the same goals for accessibility, mobility and version handling
- Text format is the only solution for all problems
- XML is ideal for text with markup information
- DocBook XML is ***the*** industry standard for documentation

DocBook XML

- Open standard
- XML based
 - Can be edited with any XML capable editor
 - Can be validated
 - Can be transformed to any output format
 - Version control is supported on the text level
- New tools are emerging for storage and editing [oc4ware]

Who uses DocBook?

- Linux Documentation Project
- KDE
- Gnome
- Apple's Darwin Documentation Project
- IBM, HP, Microsoft, O'Reilly...
- PHP
 - PHP Documentation, PEAR, PHP GK...

DocBook element examples

- Document structuring
 - <book>, <part>, <appendix>
 - <section>, <chapter>
 - <para>, <note>, <warning>
- Technical documentation
 - <reference>, <refentry>, <methodsynopsis>, <example>, <type>, <function>, <parameter>
- Cross references
 - <link>, <ulink>

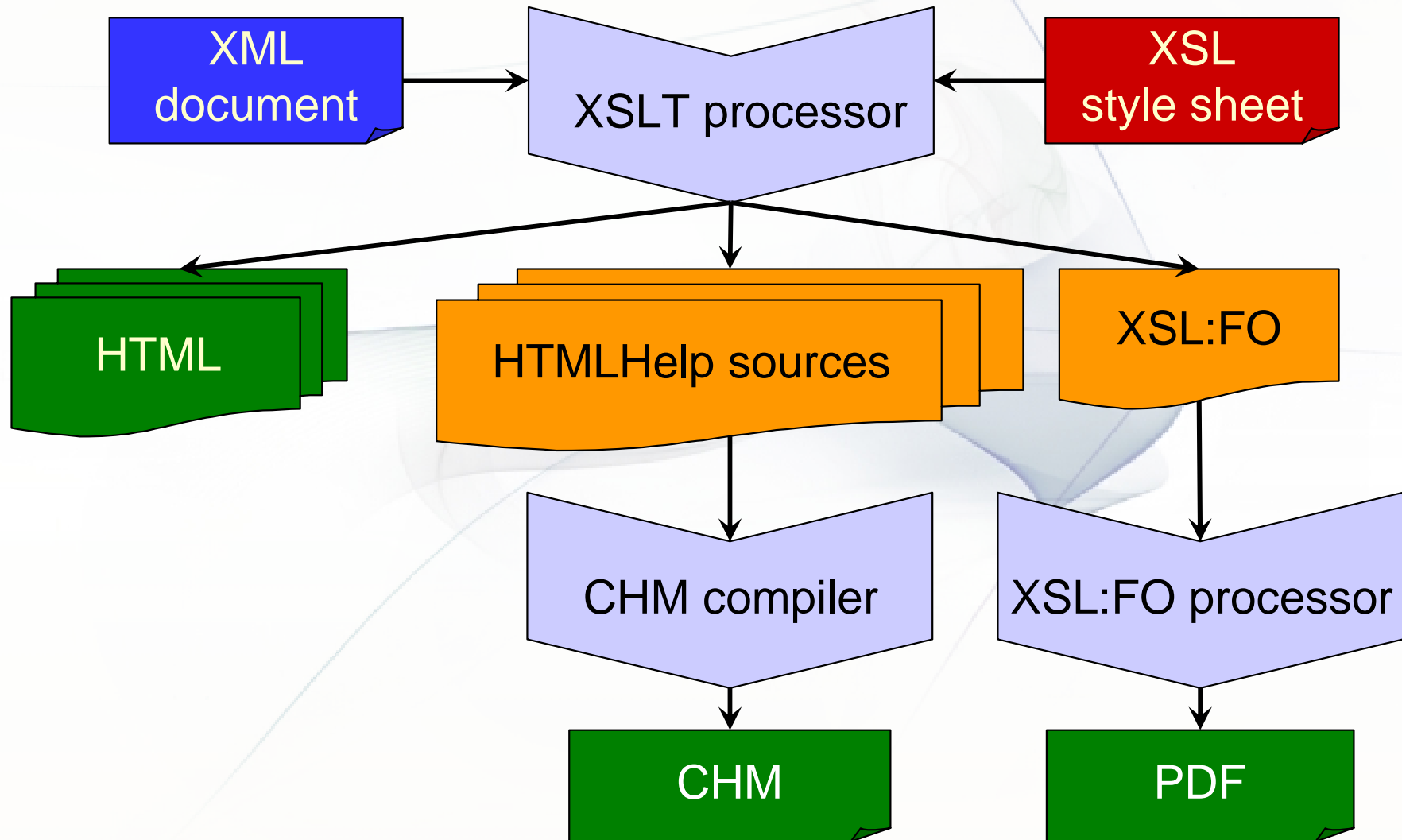
DB is not exactly what you want

- DocBook is probably not exactly what you want
- There are many tags you will never use
- It is customizable to your needs through DTD layers
 - Add / remove tags or attributes
 - Modify attributes, add default values
 - Modify content models

Creating and editing DocBook XML

- Any XML editor can edit DocBook XML documents
 - Emacs/PSGML, Vim
 - <code>oXygen</code>
 - XMLSpy ...
- Some PHP editors and other IDEs also support it on different levels
 - JEdit, Eclipse, NetBeans, PHPEdit

Transforming DocBook XML



Transforming DocBook XML

- XSLT processors
 - Java based
 - Saxon
 - Xalan (also in C++)
 - C programs- libxslt [xsltproc, xmllint]
- PDF generation
 - FOP
 - PassiveTex
 - RenderX XEP

Merging documentation

- Two different documentations
- DocBook supports API documentation too
- Convert PHPDoc comments to DocBook XML tags and content
- Two existing developments
 - Phpdoc2peardoc exclusively for PEAR
 - DocBook Doclet for Javadoc
- *A phpDocumentor converter?*

Future of documentation

- DocBook's future is guaranteed
- PHPDoc needs to be changed
 - Support for new features in PHP 5
 - Depreciation of old features in the long term
 - PEAR/PHPDoc will probably merge with phpDocumentor
 - More IDE support is needed (standard!)



Questions?



Thank you!