

Proposal for a common documentation workflow

Motivation

OSGEO intends to create a common repository for the mutual creation and edition of documentation, tutorials and teaching materials. A common format can facilitate the collaborative editing as a format and editing system known to all editors and usable by everyone regardless of the operation system, preferred editor or workflow has the following advantages:

- lower the entrance barrier for possible contributors
- reduce the number of formatting errors
- automatise the build and creation of final documents

A simple format: Restructured Text (ReST)

Restructured Text is a markup format mainly used in the documentation of software and especially adopted as standard for Python docstrings.

Advantages

- simple text format
- no styling
- widely adopted in the open-source programming world
- **converter into various export formats (latex, html, odf) exist**
 - [pandoc](#)
 - native: rst2html, rst2s5, rst2odp, rst2latex
 - additional: rst2beamer, rst2odt, rst2pdf
- easy to create screen presentations from the documents
- easy to import or include in academic publications through LaTeX/LyX

Support

- [Emacs: see docutils page](#)
- [Gedit \(Linux\)](#)
- [Ulipad \(Win\)](#)

Alternative Formats

- **Lyx which also support export to**
 - PDF
 - HTML
 - ODT

A versatile build system: Sphinx

Introduction

The documentation creation and build system [Sphinx](#) relies on ReST as input format and uses its simple structure together with the [docutils](#) conversion utilities to offer a flexible documentation system.

It is used to prepare the documentation of one of the major scripting languages: [Python](#). More and more [software projects](#) have adopted it for their documentation; and while most of them are using Python, other programming languages are also supported.

Advantages

- as being the documentation tool for Python, continuous development is ensured
- well documented
- a supportive user mailing list
- a [simple webinterface](#) for online editing available and a more advanced is [in development](#) under the current Summer of Code 2009
- editing is simple and accessible as ReST is used (see above)
- content is mostly separated from layout and styling
- theming support for layout and styling
- configurable content depending on output format
- **supports markup suited for documentaion and software tutorials such as**
 - equations
 - code highlighting
- literature citations are cared for and rudimentary bibtex plugin exists
- by consisting only of text files it is easy to include into a version control system
- the final documentation can be built into html or PDF among others
- can be build automatically with python distutils or any other build system

Examples

- **An example of a community written book using Sphinx: Python3 Patterns**
 - <http://www.mindviewinc.com/Books/Python3Patterns/Index.php>
 - <http://www.artima.com/weblogs/viewpost.jsp?thread=239183>
 - It is maintained in a DVCS: <http://bitbucket.org/BruceEckel/python-3-patterns-idioms/>