SiSU
Manual

Ralph Amissah

copy @ www.jus.uio.no/sisu/ *

* Generated by SiSU www.jus.uio.no/sisu
www.sisudoc.org

Copyright © Ralph Amissah 2007, part of SiSU documentation, License GPL 3

Generated by SiSU [SiSU 0.59.1 of 2007w39/2] www.jus.uio.no/sisu
Copyright © 1997, current 2007 Ralph Amissah, All Rights Reserved.
SiSU is software for document structuring, publishing and search (with object citation numbering), www.sisudoc.org
SiSU is released under GPL 3 or later, <http://www.fsf.org/licenses/gpl.html>.

Document information:
sourcefile sisu_manual._sst
Generated by SiSU www.jus.uio.no/sisu
version information: SiSU 0.59.1 of 2007w39/2

For alternative output formats of this document check:
<http://www.jus.uio.no/sisu/sisu_manual/sisu_manifest.html>

Contents

SiSU - Manual, Ralph Amissah 1

What is SiSU? 1
1. Introduction - What is SiSU? . . . 2
2. How does sisu work? . . . 5
3. Summary of features . . . 6
4. Help . . . 9
    4.1 SiSU Manual . . . 9
    4.2 SiSU man pages . . . 9
    4.3 SiSU built-in interactive help . . . 10
    4.4 Help Sources . . . 10
5. Commands Summary . . . 12
    5.1 Synopsis . . . 12
    5.2 Description . . . 12
    5.3 Document Processing Command Flags . . . 12
6. command line modifiers . . . 17
7. database commands . . . 18
8. Shortcuts, Shorthand for multiple flags . . . 19
9. Introduction to SiSU Markup . . . 20
    9.1 Summary . . . 20
    9.2 Markup Examples . . . 20
10. Markup of Headers . . . 22
    10.1 Sample Header . . . 22
    10.2 Available Headers . . . 22
11. Markup of Substantive Text . . . 26
    11.1 Heading Levels . . . 26
    11.2 Font Attributes . . . 26
    11.3 Indentation and bullets . . . 27
    11.4 Footnotes / Endnotes . . . 28
    11.5 Links . . . 28
    11.6 Grouped Text . . . 30
12. Composite documents markup . . . 38

Markup Syntax History 39
13. Notes related to Files-types and Markup Syntax . . . 40
14. SiSU filetypes . . . 41
    14.1 .sst .ssm .ssi marked up plain text . . . 41
    14.2 sisupod, zipped binary container (sisupod.zip, .ssp) . . . 42
15. Experimental Alternative Input Representations . . . 43
    15.1 Alternative XML . . . 43
16. Configuration . . . 45
    16.1 Determining the Current Configuration . . . 45

16.2 Configuration files (config.yml) . . . 45
17. Skins . . . 46
17.1 Document Skin . . . 46
17.2 Directory Skin . . . 46
17.3 Site Skin . . . 46
17.4 Sample Skins . . . 46
18. CSS - Cascading Style Sheets (for html, XHTML and XML) . . . 48
19. Organising Content . . . 49
19.1 Directory Structure and Mapping . . . 49
19.2 Organising Content . . . 49
20. Homepages . . . 50
20.1 Home page and other custom built pages in a sub-directory . . . 50
20.2 Home page within a skin . . . 50
21. Markup and Output Examples . . . 51
21.1 Markup examples . . . 51
21.2 A few book (and other) examples . . . 52
21.3 SQL - PostgreSQL, SQLite . . . 68
21.4 Lex Mercatoria as an example . . . 69
21.5 For good measure the markup for a document with lots of (simple) tables 69
21.6 And a link to the output of a reported case . . . 69
22. A Checklist of Output Features . . . 70
23. SiSU Search - Introduction . . . 71
24. SQL . . . 72
24.1 populating SQL type databases . . . 72
25. Postgresql . . . 73
25.1 Name . . . 73
25.2 Description . . . 73
25.3 Synopsis . . . 73
25.4 Commands . . . 73
26. Sqlite . . . 75
26.1 Name . . . 75
26.2 Description . . . 75
26.3 Synopsis . . . 75
26.4 Commands . . . 75
27. Introduction . . . 77
27.1 Search - database frontend sample, utilising database and SiSU features, including object citation numbering (backend currently PostgreSQL) . 77
27.2 Search Form . . . 77
28. Hyperestraier . . . 79
29. sisu_webrick . . . 81
29.1 Name . . . 81
29.2 Synopsis . . . 81
29.3 Description . . . 81
29.4 Summary of man page . . . 81

29.5 Document processing command flags . . . 81
29.6 Further information . . . 81
29.7 Author . . . 81
29.8 SEE ALSO . . . 82
30. Remote Source Documents . . . 83

Remote Document Output **84**
31. Remote Output . . . 85
31.1 commands . . . 85
31.2 configuration . . . 85
32. Remote Servers . . . 86

Download **87**
33. Download SiSU - Linux/Unix . . . 88
SiSU Current Version - Linux/Unix . . . 88

Installation **92**
34. Installation . . . 93
34.1 Debian . . . 93
34.2 Other Unix / Linux . . . 94
35. SiSU Components, Dependencies and Notes . . . 96
35.1 sisu . . . 96
35.2 sisu-complete . . . 97
35.3 sisu-examples . . . 98
35.4 sisu-pdf . . . 98
35.5 sisu-postgresql . . . 98
35.6 sisu-remote . . . 99
35.7 sisu-sqlite . . . 99
36. Quickstart - Getting Started Howto . . . 101
36.1 Installation . . . 101
36.2 Testing SiSU, generating output . . . 103
36.3 Getting Help . . . 104
36.4 Markup Samples . . . 104

HowTo **106**
37. Getting Help . . . 107
37.1 SiSU “man” pages . . . 107
37.2 SiSU built-in help . . . 107
37.3 Command Line with Flags - Batch Processing . . . 108
38. Setup, initialisation . . . 109
38.1 initialise output directory . . . 109
38.2 misc . . . 111
38.3 remote placement of output . . . 111
39. Configuration Files . . . 112
40. Markup . . . 113

40.1 Headers . . . 113
40.2 Font Face . . . 113
40.3 Endnotes . . . 114
40.4 Links . . . 115
40.5 Number Titles . . . 115
40.6 Line operations . . . 115
40.7 Tables . . . 116
40.8 Grouped Text . . . 117
40.9 Composite Document . . . 117
41. Change Appearance . . . 118
41.1 Skins . . . 118
41.2 CSS . . . 118

Extracts from the README **120**
42. README . . . 121
42.1 Online Information, places to look . . . 122
42.2 Installation . . . 122
42.3 Dependencies . . . 125
42.4 Quick start . . . 128
42.5 Configuration files . . . 129
42.6 Use General Overview . . . 129
42.7 Help . . . 129
42.8 Directory Structure . . . 130
42.9 Configuration File . . . 130
42.10 Markup . . . 130
42.11 Additional Things . . . 131
42.12 License . . . 131
42.13 SiSU Standard . . . 131

Extracts from man 8 sisu **132**
43. Post Installation Setup . . . 133
43.1 Post Installation Setup - Quick start . . . 133
43.2 Document markup directory . . . 134
44. FAQ - Frequently Asked/Answered Questions . . . 137
44.1 Why are urls produced with the -v (and -u) flag that point to a web server on port 8081? . . . 137
44.2 I cannot find my output, where is it? . . . 137
44.3 I do not get any pdf output, why? . . . 137
44.4 Where is the latex (or some other interim) output? . . . 137
44.5 Why isn't SiSU markup XML . . . 138
44.6 LaTeX claims to be a document preparation system for high-quality typesetting. Can the same be said about SiSU? . . . 138
44.7 How do I create GIN or GiST index in Postgresql for use in SiSU . . . 139
44.8 Where is version 1.0? . . . 140
45. Editor Files, Syntax Highlighting . . . 141

46. Help Sources 143
46.1 man pages 143
46.2 sisu generated output - links to html 144

Document Information (metadata) 150
Metadata 150

Information on this document copy and an unofficial List of Some web related information and sources 151
Information on this document copy 151
Links that may be of interest 151

SiSU - Manual, Ralph Amissah 1

What is SiSU? 2

1. Introduction - What is SiSU?

SiSU is a system for document markup, publishing (in multiple open standard formats) and search

SiSU [1] is a[2] framework for document structuring, publishing and search, comprising of (a) a lightweight document structure and presentation markup syntax and (b) an accompanying engine for generating standard document format outputs from documents prepared in sisu markup syntax, which is able to produce multiple standard outputs that (can) share a common numbering system for the citation of text within a document.

SiSU is developed under an open source, software libre license (GPL3). It has been developed in the context of coping with large document sets with evolving markup related technologies, for which you want multiple output formats, a common mechanism for cross-output-format citation, and search.

SiSU both defines a markup syntax and provides an engine that produces open standards format outputs from documents prepared with **SiSU** markup. From a single lightly prepared document sisu custom builds several standard output formats which share a common (text object) numbering system for citation of content within a document (that also has implications for search). The sisu engine works with an abstraction of the document's structure and content from which it is possible to generate different forms of representation of the document. Significantly **SiSU** markup is more sparse than html and outputs which include html, LaTeX, landscape and portrait pdfs, Open Document Format (ODF), all of which can be added to and updated. **SiSU** is also able to populate SQL type databases at an object level, which means that searches can be made with that degree of granularity. Results of objects (primarily paragraphs and headings) can be viewed directly in the database, or just the object numbers shown - your search criteria is met in these documents and at these locations within each document.

Source document preparation and output generation is a two step process: (i) document source is prepared, that is, marked up in sisu markup syntax and (ii) the desired output subsequently generated by running the sisu engine against document source. Output representations if updated (in the sisu engine) can be generated by re-running the engine against the prepared source. Using **SiSU** markup applied to a document, **SiSU** custom builds various standard open output formats including plain text, HTML, XHTML, XML, OpenDocument, LaTeX or PDF files, and populate an SQL database with objects[3] (equating generally to paragraph-sized chunks) so searches may be performed and matches returned with that degree of granularity (e.g. your search criteria is met by these documents and at these locations within each document). Document output formats share a common object numbering system for locating content. This is particularly suitable for "published" works (finalized texts as opposed to works that are frequently changed or updated) for which it provides a fixed means of reference of content.

In preparing a **SiSU** document you optionally provide semantic information related to the

[1] "**SiSU** information Structuring Universe" or "Structured information, Serialized Units".
also chosen for the meaning of the Finnish term "sisu".

[2] Unix command line oriented

[3] objects include: headings, paragraphs, verse, tables, images, but not footnotes/endnotes which are numbered separately and tied to the object from which they are referenced.

document in a document header, and in marking up the substantive text provide information on the structure of the document, primarily indicating heading levels and footnotes. You also provide information on basic text attributes where used. The rest is automatic, sisu from this information custom builds[4] the different forms of output requested.

SiSU works with an abstraction of the document based on its structure which is comprised of its frame[5] and the objects[6] it contains, which enables **SiSU** to represent the document in many different ways, and to take advantage of the strengths of different ways of presenting documents. The objects are numbered, and these numbers can be used to provide a common base for citing material within a document across the different output format types. This is significant as page numbers are not suited to the digital age, in web publishing, changing a browser's default font or using a different browser means that text appears on different pages; and in publishing in different formats, html, landscape and portrait pdf etc. again page numbers are of no use to cite text in a manner that is relevant against the different output types. Dealing with documents at an object level together with object numbering also has implications for search.

One of the challenges of maintaining documents is to keep them in a format that would allow users to use them without depending on a proprietary software popular at the time. Consider the ease of dealing with legacy proprietary formats today and what guarantee you have that old proprietary formats will remain (or can be read without proprietary software/equipment) in 15 years time, or the way the way in which html has evolved over its relatively short span of existence. **SiSU** provides the flexibility of outputing documents in multiple non-proprietary open formats including html, pdf[7] and the ISO standard ODF.[8] Whilst **SiSU** relies on software, the markup is uncomplicated and minimalistic which guarantees that future engines can be written to run against it. It is also easily converted to other formats, which means documents prepared in **SiSU** can be migrated to other document formats. Further security is provided by the fact that the software itself, **SiSU** is available under GPL3 a licence that guarantees that the source code will always be open, and free as in libre which means that that code base can be used updated and further developed as required under the terms of its license. Another challenge is to keep up with a moving target. **SiSU** permits new forms of output to be added as they become important, (Open Document Format text was added in 2006), and existing output to be updated (html has evolved and the related module has been updated repeatedly over the years, presumably when the World Wide Web Consortium (w3c) finalises html 5 which is currently under development, the html module will again be updated allowing all existing documents to be regenerated as html 5).

The document formats are written to the file-system and available for indexing by independent indexing tools, whether off the web like Google and Yahoo or on the site like Lucene and Hyperestraier.

SiSU also provides other features such as concordance files and document content certificates,

[4] i.e. the html, pdf, odf outputs are each built individually and optimised for that form of presentation, rather than for example the html being a saved version of the odf, or the pdf being a saved version of the html.

[5] the different heading levels

[6] units of text, primarily paragraphs and headings, also any tables, poems, code-blocks

[7] Specification submitted by Adobe to ISO to become a full open ISO specification <http://www.linux-watch.com/news/NS7542722606.html>

[8] ISO/IEC 26300:2006

and the working against an abstraction of document structure has further possibilities for the research and development of other document representations, the availability of objects is useful for example for topic maps and the commercial law thesaurus by Vikki Rogers and Al Krtizer, together with the flexibility of **SiSU** offers great possibilities.

SiSU is primarily for published works, which can take advantage of the citation system to reliably reference its documents. **SiSU** works well in a complementary manner with such collaborative technologies as Wikis, which can take advantage of and be used to discuss the substance of content prepared in **SiSU** .

<http://www.jus.uio.no/sisu>

2. How does sisu work?

SiSU markup is fairly minimalistic, it consists of: a (largely optional) document header, made up of information about the document (such as when it was published, who authored it, and granting what rights) and any processing instructions; and markup within the substantive text of the document, which is related to document structure and typeface. **SiSU** must be able to discern the structure of a document, (text headings and their levels in relation to each other), either from information provided in the document header or from markup within the text (or from a combination of both). Processing is done against an abstraction of the document comprising of information on the document's structure and its objects,[2] which the program serializes (providing the object numbers) and which are assigned hash sum values based on their content. This abstraction of information about document structure, objects, (and hash sums), provides considerable flexibility in representing documents different ways and for different purposes (e.g. search, document layout, publishing, content certification, concordance etc.), and makes it possible to take advantage of some of the strengths of established ways of representing documents, (or indeed to create new ones).

3. Summary of features

- sparse/minimal markup (clean utf-8 source texts). Documents are prepared in a single UTF-8 file using a minimalistic mnemonic syntax. Typical literature, documents like "War and Peace" require almost no markup, and most of the headers are optional.

- markup is easily readable/parsable by the human eye, (basic markup is simpler and more sparse than the most basic HTML), [this may also be converted to XML representations of the same input/source document].

- markup defines document structure (this may be done once in a header pattern-match description, or for heading levels individually); basic text attributes (bold, italics, underscore, strike-through etc.) as required; and semantic information related to the document (header information, extended beyond the Dublin core and easily further extended as required); the headers may also contain processing instructions. **SiSU** markup is primarily an abstraction of document structure and document metadata to permit taking advantage of the basic strengths of existing alternative practical standard ways of representing documents [be that browser viewing, paper publication, sql search etc.] (html, xml, odf, latex, pdf, sql)

- for output produces reasonably elegant output of established industry and institutionally accepted open standard formats.[3] takes advantage of the different strengths of various standard formats for representing documents, amongst the output formats currently supported are:

    - html - both as a single scrollable text and a segmented document
    - xhtml
    - XML - both in sax and dom style xml structures for further development as required
    - ODF - open document format, the iso standard for document storage
    - LaTeX - used to generate pdf
    - pdf (via LaTeX)
    - sql - population of an sql database, (at the same object level that is used to cite text within a document)

Also produces: concordance files; document content certificates (md5 or sha256 digests of headings, paragraphs, images etc.) and html manifests (and sitemaps of content). (b) takes advantage of the strengths implicit in these very different output types, (e.g. PDFs produced using typesetting of LaTeX, databases populated with documents at an individual object/paragraph level, making possible granular search (and related possibilities))

- ensuring content can be cited in a meaningful way regardless of selected output format. Online publishing (and publishing in multiple document formats) lacks a useful way of citing text internally within documents (important to academics generally and to lawyers) as page numbers are meaningless across browsers and formats. sisu seeks to provide a common way of pinpoint the text within a document, (which can be utilized for citation and by search engines). The outputs share a common numbering system that is meaningful (to man and machine) across

all digital outputs whether paper, screen, or database oriented, (pdf, HTML, xml, sqlite, postgresql), this numbering system can be used to reference content.

• Granular search within documents. SQL databases are populated at an object level (roughly headings, paragraphs, verse, tables) and become searchable with that degree of granularity, the output information provides the object/paragraph numbers which are relevant across all generated outputs; it is also possible to look at just the matching paragraphs of the documents in the database; [output indexing also work well with search indexing tools like hyperestraier]. 32

• long term maintainability of document collections in a world of changing formats, having a very sparsely marked-up source document base. there is a considerable degree of future-proofing, output representations are "upgradeable", and new document formats may be added. e.g. addition of odf (open document text) module in 2006 and in future html5 output sometime in future, without modification of existing prepared texts 33

• SQL search aside, documents are generated as required and static once generated. 34

• documents produced are static files, and may be batch processed, this needs to be done only once but may be repeated for various reasons as desired (updated content, addition of new output formats, updated technology document presentations/representations) 35

• document source (plaintext utf-8) if shared on the net may be used as input and processed locally to produce the different document outputs 36

• document source may be bundled together (automatically) with associated documents (multiple language versions or master document with inclusions) and images and sent as a zip file called a sisupod, if shared on the net these too may be processed locally to produce the desired document outputs 37

• generated document outputs may automatically be posted to remote sites. 38

• for basic document generation, the only software dependency is **Ruby** , and a few standard Unix tools (this covers plaintext, HTML, XML, ODF, LaTeX). To use a database you of course need that, and to convert the LaTeX generated to pdf, a latex processor like tetex or texlive. 39

• as a developers tool it is flexible and extensible 40

Syntax highlighting for **SiSU** markup is available for a number of text editors. 41

SiSU is less about document layout than about finding a way with little markup to be able to construct an abstract representation of a document that makes it possible to produce multiple representations of it which may be rather different from each other and used for different purposes, whether layout and publishing, or search of content 42

i.e. to be able to take advantage from this minimal preparation starting point of some of the strengths of rather different established ways of representing documents for different purposes, whether for search (relational database, or indexed flat files generated for that purpose whether of complete documents, or say of files made up of objects), online viewing (e.g. html, xml, pdf), or paper publication (e.g. pdf)... 43

the solution arrived at is by extracting structural information about the document (about headings within the document) and by tracking objects (which are serialized and also given hash values) in the manner described. It makes possible representations that are quite different from those offered at present. For example objects could be saved individually and identified by their hashes, with an index of how the objects relate to each other to form a document.

4. Help

4.1 SiSU Manual

The most up to date information on sisu should be contained in the sisu_manual, available at:

<http://www.jus.uio.no/sisu/sisu_manual/>

and (from **SiSU** 0.59 onwards) installed locally at:

/usr/share/doc/sisu/sisu_manual/

or equivalent directory

Within the **SiSU** tarball at:

./data/doc/sisu/sisu_manual/

4.2 SiSU man pages

If **SiSU** is installed on your system usual man commands should be available, try:

man sisu

man sisu_markup

man sisu_commands

Most **SiSU** man pages are generated directly from sisu documents that are used to prepare the sisu manual, the sources files for which are located within the **SiSU** tarball at:

./data/doc/sisu/sisu_manual/

Once installed, directory equivalent to:

/usr/share/doc/sisu/sisu_manual/

Available man pages are converted back to html using man2html:

/usr/share/doc/sisu/html/

./data/doc/sisu/html/

The **SiSU** man pages can be viewed online at:[9]

An online version of the sisu man page is available here:

- various sisu man pages [10]

[9] generated from source using rman
<http://polyglotman.sourceforge.net/rman.html>
With regard to **SiSU** man pages the formatting generated for markup syntax is not quite right, for that you might prefer the links under:
<http://www.jus.uio.no/sample>

- sisu.1 [11]
- sisu.8 [12]
- sisu_examples.1 [13]
- sisu_webrick.1 [14]

4.3 SiSU built-in interactive help

This is particularly useful when current installation information is obtained as the interactive help is able to provide information on your sisu configuration and setup.

sisu –help

sisu –help [subject]

sisu –help env [for feedback on the way your system is setup with regard to sisu]

sisu -V [same as above command]

sisu –help commands

sisu –help markup

Apart from real-time information on your current configuration the **SiSU** manual and man pages are likely to contain more up-to-date information than the sisu interactive help (for example on commands and markup).

NOTE: Running the command sisu (alone without any flags, filenames or wildcards) brings up the interactive help, as does any sisu command that is not recognised. Enter to escape.

4.4 Help Sources

For lists of alternative help sources, see:

man page

man sisu_help_sources

man2html

/usr/share/doc/sisu/sisu_manual/sisu_help_sources/index.html

sisu generated html

/usr/share/doc/sisu/html/sisu_help_sources/index.html

[10] <http://www.jus.uio.no/sisu/man/>
[11] <http://www.jus.uio.no/sisu/man/sisu.1.html>
[12] <http://www.jus.uio.no/sisu/man/sisu.8.html>
[13] <http://www.jus.uio.no/sisu/man/sisu_examples.1.html>
[14] <http://www.jus.uio.no/sisu/man/sisu_webrick.1.html>

＜http://sisudoc.org/sisu_manual/sisu_help_sources/index.html＞ 91

＜http://www.jus.uio.no/sisu/sisu_manual/sisu_help_sources/index.html＞ 92

5. Commands Summary

5.1 Synopsis

SiSU - Structured information, Serialized Units - a document publishing system

sisu [-abcDdFHhIiMmNnopqRrSsTtUuVvwXxYyZz0-9] [filename/ wildcard]

sisu [-Ddcv] [instruction]

sisu [-CcFLSVvW]

Note: commands should be issued from within the directory that contains the marked up files, cd to markup directory.

5.2 Description

SiSU **SiSU** is a document publishing system, that from a simple single marked-up document, produces multiple of output formats including: plaintext, html, LaTeX, pdf, xhtml, XML, info, and SQL (PostgreSQL and SQLite), which share numbered text objects ("object citation numbering") and the same document structure information. For more see: <http://www.jus.uio.no/sisu>

5.3 Document Processing Command Flags

-a [filename/wildcard]
produces plaintext with Unix linefeeds and without markup, (object numbers are omitted), has footnotes at end of each paragraph that contains them [-A for equivalent dos (linefeed) output file] [see -e for endnotes]. (Options include: –endnotes for endnotes –footnotes for footnotes at the end of each paragraph –unix for unix linefeed (default) –msdos for msdos linefeed)

-b [filename/wildcard]
produces xhtml/XML output for browser viewing (sax parsing).

-C [–init-site]
configure/initialise shared output directory files initialize shared output directory (config files such as css and dtd files are not updated if they already exist unless modifier is used). -C –init-site configure/initialise site more extensive than -C on its own, shared output directory files/force update, existing shared output config files such as css and dtd files are updated if this modifier is used.

-CC
configure/initialise shared output directory files initialize shared output directory (config files such as css and dtd files are not updated if they already exist unless modifier is used). The equivalent of: -C –init-site configure/initialise site, more extensive than -C on its own, shared output directory files/force update, existing shared output config files such as css and dtd files are updated if -CC is used.

-c [filename/wildcard] 107
screen toggle ansi screen colour on or off depending on default set (unless -c flag is used: if sisurc colour default is set to 'true', output to screen will be with colour, if sisurc colour default is set to 'false' or is undefined screen output will be without colour).

-D [instruction] [filename] 108
database postgresql (–pgsql may be used instead) possible instructions, include: –createdb; –create; –dropall; –import [filename]; –update [filename]; –remove [filename]; see database section below.

-d [–db-[database type (sqlite|pg)]] –[instruction] [filename] 109
database type default set to sqlite, (for which –sqlite may be used instead) or to specify another database –db-[pgsql, sqlite] (however see -D) possible instructions include: –createdb; –create; –dropall; –import [filename]; –update [filename]; –remove [filename]; see database section below.

-F [–webserv=webrick] 110
generate examples of (naive) cgi search form for sqlite and pgsql depends on your already having used sisu to populate an sqlite and/or pgsql database, (the sqlite version scans the output directories for existing sisu_sqlite databases, so it is first necessary to create them, before generating the search form) see -d -D and the database section below. If the optional parameter –webserv=webrick is passed, the cgi examples created will be set up to use the default port set for use by the webrick server, (otherwise the port is left blank and the system setting used, usually 80). The samples are dumped in the present work directory which must be writable, (with screen instructions given that they be copied to the cgi-bin directory). -Fv (in addition to the above) provides some information on setting up hyperestraier for sisu

-H [filename/wildcard] 111
produces html without link suffixes (.html .pdf etc.) ("Hide"). Requires an appropriately configured web server. [behaviour switched after 0.35 see -h].

-h [filename/wildcard] 112
produces html (with hardlinks i.e. with name suffixes in links/local urls). html, with internal document links that include the document suffix, i.e. whether it is .html or .pdf (required for browsing directly off a file system, and works with most web servers). [behaviour switched after 0.35 see -H].

-I [filename/wildcard] 113
produces texinfo and info file, (view with pinfo).

-L 114
prints license information.

-M [filename/wildcard/url] 115
maintenance mode files created for processing preserved and their locations indicated. (also see -V)

-m [filename/wildcard/url] 116
assumed for most other flags, creates new meta-markup file, (the metaverse) that is used in all

subsequent processing of other output. This step is assumed for most processing flags. To skip it see -n

-N [filename/wildcard/url] 117
document digest or document content certificate (DCC) as md5 digest tree of the document: the digest for the document, and digests for each object contained within the document (together with information on software versions that produced it) (digest.txt). -NV for verbose digest output to screen.

-n [filename/wildcard/url] 118
skip meta-markup (building of "metaverse"), this skips the equivalent of -m which is otherwise assumed by most processing flags.

-o [filename/wildcard/url] 119
output basic document in opendocument file format (opendocument.odt).

-p [filename/wildcard] 120
produces LaTeX pdf (portrait.pdf and landscape.pdf). Default paper size is set in config file, or document header, or provided with additional command line parameter, e.g. –papersize-a4 preset sizes include: 'A4', U.S. 'letter' and 'legal' and book sizes 'A5' and 'B5' (system defaults to A4).

-q [filename/wildcard] 121
quiet less output to screen.

-R [filename/wildcard] 122
copies sisu output files to remote host using rsync. This requires that sisurc.yml has been provided with information on hostname and username, and that you have your "keys" and ssh agent in place. Note the behavior of rsync different if -R is used with other flags from if used alone. Alone the rsync –delete parameter is sent, useful for cleaning the remote directory (when -R is used together with other flags, it is not). Also see -r

-r [filename/wildcard] 123
copies sisu output files to remote host using scp. This requires that sisurc.yml has been provided with information on hostname and username, and that you have your "keys" and ssh agent in place. Also see -R

-S 124
produces a sisupod a zipped sisu directory of markup files including sisu markup source files and the directories local configuration file, images and skins. Note: this only includes the configuration files or skins contained in ./_sisu not those in ~/.sisu -S [filename/wildcard] option. Note: (this option is tested only with zsh).

-S [filename/wildcard] 125
produces a zipped file of the prepared document specified along with associated images, by default named sisupod.zip they may alternatively be named with the filename extension .ssp This provides a quick way of gathering the relevant parts of a sisu document which can then for example be emailed. A sisupod includes sisu markup source file, (along with associated documents if a master file, or available in multilingual versions), together with related images

and skin. **SiSU** commands can be run directly against a sisupod contained in a local directory, or provided as a url on a remote site. As there is a security issue with skins provided by other users, they are not applied unless the flag –trust or –trusted is added to the command instruction, it is recommended that file that are not your own are treated as untrusted. The directory structure of the unzipped file is understood by sisu, and sisu commands can be run within it. Note: if you wish to send multiple files, it quickly becomes more space efficient to zip the sisu markup directory, rather than the individual files for sending). See the -S option without [filename/wildcard].

-s [filename/wildcard]
copies sisu markup file to output directory.

-t [filename/wildcard (*.termsheet.rb)]
standard form document builder, preprocessing feature

-U [filename/wildcard]
prints url output list/map for the available processing flags options and resulting files that could be requested, (can be used to get a list of processing options in relation to a file, together with information on the output that would be produced), -u provides url output mapping for those flags requested for processing. The default assumes sisu_webrick is running and provides webrick url mappings where appropriate, but these can be switched to file system paths in sisurc.yml

-u [filename/wildcard]
provides url mapping of output files for the flags requested for processing, also see -U

-V
on its own, provides **SiSU** version and environment information (sisu –help env)

-V [filename/wildcard]
even more verbose than the -v flag. (also see -M)

-v
on its own, provides **SiSU** version information

-v [filename/wildcard]
provides verbose output of what is being built, where it is being built (and error messages if any), as with -u flag provides a url mapping of files created for each of the processing flag requests. See also -V

-W
starts ruby's webrick webserver points at sisu output directories, the default port is set to 8081 and can be changed in the resource configuration files. [tip: the webrick server requires link suffixes, so html output should be created using the -h option rather than -H; also, note -F webrick].

-w [filename/wildcard]
produces concordance (wordmap) a rudimentary index of all the words in a document. (Concordance files are not generated for documents of over 260,000 words unless this limit is increased

in the file sisurc.yml)

-X [filename/wildcard]
produces XML output with deep document structure, in the nature of dom.

-x [filename/wildcard]
produces XML output shallow structure (sax parsing).

-Y [filename/wildcard]
produces a short sitemap entry for the document, based on html output and the sisu_manifest. –sitemaps generates/updates the sitemap index of existing sitemaps. (Experimental, [g,y,m announcement this week])

-y [filename/wildcard]
produces an html summary of output generated (hyperlinked to content) and document specific metadata (sisu_manifest.html). This step is assumed for most processing flags.

-Z [filename/wildcard]
Zap, if used with other processing flags deletes output files of the type about to be processed, prior to processing. If -Z is used as the lone processing related flag (or in conjunction with a combination of -[mMvVq]), will remove the related document output directory.

-z [filename/wildcard]
produces php (zend) [this feature is disabled for the time being]

6. command line modifiers 142

–no-ocn 143

with -h -H or -p

switches off object citation numbering. Produce output without identifying numbers in margins of html or LaTeX/pdf output.

–no-annotate 144

strips output text of editor endnotes[*1] denoted by asterisk or dagger/plus sign

–no-asterisk 145

strips output text of editor endnotes[*2] denoted by asterisk sign

–no-dagger 146

strips output text of editor endnotes[+1] denoted by dagger/plus sign

[*1] square brackets
[*2] square brackets
[+1] square brackets

7. database commands

dbi - database interface

-D or –pgsql set for postgresql -d or –sqlite default set for sqlite -d is modifiable with –db=[database type (pgsql or sqlite)]

-Dv –createall
initial step, creates required relations (tables, indexes) in existing postgresql database (a database should be created manually and given the same name as working directory, as requested) (rb.dbi) [-dv –createall sqlite equivalent] it may be necessary to run sisu -Dv –createdb initially NOTE: at the present time for postgresql it may be necessary to manually create the database. The command would be 'createdb [database name]' where database name would be SiSU_[present working directory name (without path)]. Please use only alphanumerics and underscores.

-Dv –import
filename/wildcard
imports data specified to postgresql db (rb.dbi) [-dv –import sqlite equivalent]

-Dv –update
filename/wildcard
updates/imports specified data to postgresql db (rb.dbi) [-dv –update sqlite equivalent]

-D –remove
filename/wildcard
removes specified data to postgresql db (rb.dbi) [-d –remove sqlite equivalent]

-D –dropall
kills data" and drops (postgresql or sqlite) db, tables and indexes [-d –dropall sqlite equivalent]

The v in e.g. -Dv is for verbose output.

8. Shortcuts, Shorthand for multiple flags

–update [filename/wildcard]

Checks existing file output and runs the flags required to update this output. This means that if only html and pdf output was requested on previous runs, only the -hp files will be applied, and only these will be generated this time, together with the summary. This can be very convenient, if you offer different outputs of different files, and just want to do the same again.

-0 to -5 [filename or wildcard]

Default shorthand mappings (note that the defaults can be changed/configured in the sisurc.yml file):

-0

-mNhwpAobxXyYv [this is the default action run when no options are give, i.e. on 'sisu [filename]']

-1

-mNHwpy

-2

-mNHwpaoy

-3

-mNhwpAobxXyY

-4

-mNhwpAobxXDyY –import

-5

-mNhwpAobxXDyY –update

add -v for verbose mode and -c for color, e.g. sisu -2vc [filename or wildcard]

consider -u for appended url info or -v for verbose output

8.0.1 Command Line with Flags - Batch Processing

In the data directory run sisu -mh filename or wildcard eg. "sisu -h cisg.sst" or "sisu -h *.{sst,ssm}" to produce html version of all documents.

Running sisu (alone without any flags, filenames or wildcards) brings up the interactive help, as does any sisu command that is not recognised. Enter to escape.

9. Introduction to SiSU Markup[15]

9.1 Summary

SiSU source documents are plaintext (UTF-8)[16] files

All paragraphs are separated by an empty line.

Markup is comprised of:

- at the top of a document, the document header made up of semantic meta-data about the document and if desired additional processing instructions (such an instruction to automatically number headings from a particular level down)
- followed by the prepared substantive text of which the most important single characteristic is the markup of different heading levels, which define the primary outline of the document structure. Markup of substantive text includes:
    - heading levels defines document structure
    - text basic attributes, italics, bold etc.
    - grouped text (objects), which are to be treated differently, such as code blocks or poems.
    - footnotes/endnotes
    - linked text and images
    - paragraph actions, such as indent, bulleted, numbered-lists, etc.

Some interactive help on markup is available, by typing sisu and selecting markup or sisu –help markup

9.2 Markup Examples

9.2.1 Online

Online markup examples are available together with the respective outputs produced from <http://www.jus.uio.no/sisu/SiSU/2.html> or from <http://www.jus.uio.no/sisu/sisu_examples/>

There is of course this document, which provides a cursory overview of sisu markup and the respective output produced: <http://www.jus.uio.no/sisu/sisu_markup/>

Some example marked up files are available as html with syntax highlighting for viewing: <http://www.jus.uio.no/sisu/sample/syntax>

an alternative presentation of markup syntax: <http://www.jus.uio.no/sisu/sample/on_markup.txt>

[15] From sometime after SiSU 0.58 it should be possible to describe SiSU markup using SiSU, which though not an original design goal is useful.

[16] files should be prepared using UTF-8 character encoding

9.2.2 Installed

With **SiSU** installed sample skins may be found in: /usr/share/doc/sisu/sisu_markup_samples/dfsg
(or equivalent directory) and if sisu-markup-samples is installed also under: /usr/share/doc/sisu/sisu_markup_sa
free

10. Markup of Headers

Headers consist of semantic meta-data about a document, which can be used by any output module of the program; and may in addition include extra processing instructions.

Note: the first line of a document may include information on the markup version used in the form of a comment. Comments are a percentage mark at the start of a paragraph (and as the first character in a line of text) followed by a space and the comment:

```
% this would be a comment
```

10.1 Sample Header

This current document has a header similar to this one (without the comments):

```
% SiSU 0.57
@title: SiSU
@subtitle: Markup
@creator: Ralph Amissah
@rights: Copyright (C) Ralph Amissah 2007, part of SiSU documentation, License GPL 3
@type: information
@subject: ebook, epublishing, electronic book, electronic publishing, electronic document, electronic citation, data structure, citation systems, search
@date.created: 2002-08-28
@date.issued: 2002-08-28
@date.available: 2002-08-28
@date.modified: 2007-09-16
@date: 2007-09-16
@level: new=C; break=1; num_top=1
% comment: in this @level header num_top=1 starts automatic heading numbering at heading level 1 (numbering continues 3 levels down); the new and break instructions are used by the LaTeX/pdf and odf output to determine where to put page breaks (that are not used by html output or say sql database population).
@skin: skin_sisu_manual
% skins modify the appearance of a document and are placed in a sub-directory under ./_sisu/skin ~ /.sisu/skin or /etc/sisu/skin. A skin may affect single documents that request them, all documents in a directory, or be site-wide. (A document is affected by a single skin)
@bold: /Gnu|Debian|Ruby|SiSU/
@links: { SiSU Manual }http://www.jus.uio.no/sisu/sisu_manual/
{ Book Samples and Markup Examples }http://www.jus.uio.no/sisu/SiSU/2.html
{ SiSU @ Wikipedia }http://en.wikipedia.org/wiki/SiSU
{ SiSU @ Freshmeat }http://freshmeat.net/projects/sisu/
{ SiSU @ Ruby Application Archive }http://raa.ruby-lang.org/project/sisu/
{ SiSU @ Debian }http://packages.qa.debian.org/s/sisu.html
{ SiSU Download }http://www.jus.uio.no/sisu/SiSU/download.html
{ SiSU Changelog }http://www.jus.uio.no/sisu/SiSU/changelog.html
```

10.2 Available Headers

Header tags appear at the beginning of a document and provide meta information on the document (such as the Dublin Core), or information as to how the document as a whole is to be processed. All header instructions take either the form @headername: or 0~ headername. All Dublin Core meta tags are available

@indentifier: information or instructions

where the “identifier” is a tag recognised by the program, and the “information” or “instructions” belong to the tag/indentifier specified

Note: a header where used should only be used once; all headers apart from @title: are optional; the @structure: header is used to describe document structure, and can be useful to know.

This is a sample header

% SiSU 0.38 [declared file-type identifier with markup version]

@title: [title text] This is the title of the document and used as such, this header is the only one that is mandatory

@subtitle: The Subtitle if any

@creator: [or @author:] Name of Author

@subject: (whatever your subject)

@description:

@publisher:

@contributor:

@translator: [or @translated_by:]

@illustrator: [or @illustrated_by:]

@prepared_by: [or @digitized_by:]

@date: 2000-08-27 [also @date.created: @date.issued: @date.available: @date.valid: @date.modified:]

@type: article

@format:

@identifier:

@source:

@language: [or @language.document:] [country code for language if available, or language, English, en is the default setting] (en - English, fr - French, de - German, it - Italian, es - Spanish, pt - Portuguese, sv - Swedish, da - Danish, fi - Finnish, no - Norwegian, is - Icelandic, nl - Dutch, et - Estonian, hu - Hungarian, pl - Polish, ro - Romanian, ru - Russian, el - Greek, uk - Ukranian, tr - Turkish, sk - Slovak, sl - Slovenian, hr - Croatian, cs - Czech, bg - Bul garian) [however, encodings are not available for all of the languages listed.]

[@language.original: original language in which the work was published]

@papersize: (A4|US_letter|book_B5|book_A5|US_legal)

@relation:

@coverage:

@rights: Copyright (c) Name of Right Holder, all rights reserved, or as granted: public domain, copyleft, creative commons variant, etc.

@owner:

@keywords: text document generation processing management latex pdf structured xml citation [your keywords here, used for example by rss feeds, and in sql searches]

@abstract: [paper abstract, placed after table of contents]

@comment: [...]

@catalogue: loc=[Library of Congress classification]; dewey=[Dewey classification]; isbn=[ISBN]; pg=[Project Gutenberg text number]

@classify_loc: [Library of Congress classification]

@classify_dewey: [Dewey classification]

@classify_isbn: [ISBN]

@classify_pg: [Project Gutenberg text number]

@prefix: [prefix is placed just after table of contents]

@prefix_a: [prefix is placed just before table of contents - not implemented]

@prefix_b:

@rcs: $Id: sisu_markup.sst,v 1.2 2007/09/08 17:12:47 ralph Exp $ [used by rcs or cvs to embed version (revision control) information into document, rcs or cvs can usefully provide a history of updates to a document]

@structure: PART; CHAPTER; SECTION; ARTICLE; none; none;
optional, document structure can be defined by words to match or regular expression (the regular expression is assumed to start at the beginning of a line of text i.e. $)defaultmarkers : A\tilde{}to : C\tilde{}and1\tilde{}o6\check{c}anbeusedwithintextinstead, withoutthisheadertag, andmaybeusedtosupplementtheinstructi$

@level: newpage=3; breakpage=4
paragraph level, used by latex to breakpages, the page is optional eg. in newpage

@markup: information on the markup used, e.g. new=1,2,3; break=4; num_top=4 [or newpage=1,2,3; breakpage=4; num_top=4] newpage and breakpage, heading level, used by LaTeX to breakpages. breakpage: starts on a new page in single column text and on a new column in double column text; newpage: starts on a new page for both single and double column texts.
num_top=4 [auto-number document, starting at level 4. the default is to provide 3 levels, as in 1 level 4, 1.1 level 5, 1.1.1 level 6, markup to be merged within level]
num_extract [take numbering of headings provided (manually in marked up source document), and use for numbering of segments. Available where a clear numbering structure is provided within document, without the repetition of a number in a header.] [In 0.38 notation, you would map to the equivalent levels, the examples provided would map to the following new=A,B,C; break=1; num_top=1 [or newpage=A,B,C; breakpage=1; num_top=1] see headings]

@bold: [regular expression of words/phrases to be made bold]

@italics: [regular expression of words/phrases to italicise]

@vocabulary: name of taxonomy/vocabulary/wordlist to use against document

@skin: skin_doc_[name_of_desired_document_skin]

skins change default settings related to the appearance of documents generated, such as the urls of the home site, and the icon/logo for the document or site.

@links: { **SiSU** }http://www.jus.uio.no/sisu/; 247
{ FSF }http://www.fsf.org

@promo: sisu, ruby, search_libre_docs, open_society 248

places content in right pane in html, makes use of list.yml and promo.yml, commented out sample in document sample: free_as_in_freedom.richard_stallman_crusade_for_free_software.sam_williams.sst

11. Markup of Substantive Text

11.1 Heading Levels

Heading levels are :A~ ,:B~ ,:C~ ,1~ ,2~ ,3~ ... :A - :C being part / section headings, followed by other heading levels, and 1 -6 being headings followed by substantive text or sub-headings. :A~ usually the title :A~ ? conditional level 1 heading (used where a stand-alone document may be imported into another)

:A~ [heading text] Top level heading [this usually has similar content to the title @title:] NOTE: the heading levels described here are in 0.38 notation, see heading

:B~ [heading text] Second level heading [this is a heading level divider]

:C~ [heading text] Third level heading [this is a heading level divider]

1~ [heading text] Top level heading preceding substantive text of document or sub-heading 2, the heading level that would normally be marked 1. or 2. or 3. etc. in a document, and the level on which sisu by default would break html output into named segments, names are provided automatically if none are given (a number), otherwise takes the form 1~my_filename_for_this_segment

2~ [heading text] Second level heading preceding substantive text of document or sub-heading 3, the heading level that would normally be marked 1.1 or 1.2 or 1.3 or 2.1 etc. in a document.

3~ [heading text] Third level heading preceding substantive text of document, that would normally be marked 1.1.1 or 1.1.2 or 1.2.1 or 2.1.1 etc. in a document

1~ filename level 1 heading,

% the primary division such as Chapter that is followed by substantive text, and may be further subdivided (this is the level on which by default html segments are made)

11.2 Font Attributes

markup example:

normal text !{emphasis}! *{bold text}* _{underscore}_ /{italics}/ "{citation}" {*superscript*}'{*subscript*}, +{*insertedtext*}+

```
normal text
!{emphasis}!
*{bold text}*
_{underscore}_
/{italics}/
"{citation}"
{superscript}
,{subscript},
+{inserted text}+
-{strikethrough}-
```

resulting output:

normal text **emphasis** **bold text** underscore *italics* ¡cite¿citation¡/cite¿ superscript subscript inserted text ~~strikethrough~~

normal text

emphasis

bold text

underscore

italics

¡cite¿citation¡/cite¿

superscript

subscript

inserted text

~~strikethrough~~

11.3 Indentation and bullets

markup example:

```
ordinary paragraph
_1 indent paragraph one step
_2 indent paragraph two steps
_9 indent paragraph nine steps
```

resulting output:

ordinary paragraph

indent paragraph one step

indent paragraph two steps

indent paragraph nine steps

markup example:

```
_* bullet text
_1* bullet text, first indent
_2* bullet text, two step indent
```

resulting output:

- bullet text
  - bullet text, first indent
    - bullet text, two step indent

Numbered List (not to be confused with headings/titles, (document structure))

markup example:

```
# numbered list           numbered list 1., 2., 3, etc.
<=hash¿ numbered list numbered list indented a., b., c., d., etc.
```

11.4 Footnotes / Endnotes

Footnotes and endnotes not distinguished in markup. They are automatically numbered. Depending on the output file format (html, odf, pdf etc.), the document output selected will have either footnotes or endnotes.

markup example:

```
~{ a footnote or endnote }
```

resulting output:

17

markup example:

```
normal text~{ self contained endnote marker & endnote in one }~continues
```

resulting output:

normal text[18] continues

markup example:

```
normal text ~{* unnumbered asterisk footnote/endnote, insert multiple asterisks if required }~continues
normal text ~{** another unnumbered asterisk footnote/endnote }~continues
```

resulting output:

normal text [*] continues

normal text [**] continues

markup example:

```
normal text ~[* editors notes, numbered asterisk footnote/endnote series ]~continues
normal text ~[+ editors notes, numbered asterisk footnote/endnote series ]~continues
```

resulting output:

normal text [*3] continues

normal text [+2] continues

Alternative endnote pair notation for footnotes/endnotes:

```
% note the endnote marker "
```

the standard and pair notation cannot be mixed in the same document

11.5 Links

11.5.1 Naked URLs within text, dealing with urls

urls are found within text and marked up automatically. A url within text is automatically

[17] a footnote or endnote

[18] self contained endnote marker and endnote in one

[*] unnumbered asterisk footnote/endnote, insert multiple asterisks if required

[**] another unnumbered asterisk footnote/endnote

[*3] editors notes, numbered asterisk footnote/endnote series

[+2] editors notes, numbered asterisk footnote/endnote series

hyperlinked to itself and by default decorated with angled braces, unless they are contained within a code block (in which case they are passed as normal text), or escaped by a preceding underscore (in which case the decoration is omitted).

markup example:

```
normal text http://www.jus.uio.no/sisu continues
```

resulting output:

normal text <http://www.jus.uio.no/sisu> continues

An escaped url without decoration

markup example:

```
normal text http://www.jus.uio.no/sisu continues
deb http://www.jus.uio.no/sisu/archive unstable main non-free
```

resulting output:

normal text http://www.jus.uio.no/sisu continues

deb http://www.jus.uio.no/sisu/archive unstable main non-free

where a code block is used there is neither decoration nor hyperlinking, code blocks are discussed later in this document

resulting output:

```
deb http://www.jus.uio.no/sisu/archive unstable main non-free
deb-src http://www.jus.uio.no/sisu/archive unstable main non-free
```

To link text or an image to a url the markup is as follows

markup example:

```
about { SiSU }http://url.org markup
```

11.5.2 Linking Text

resulting output:

about SiSU markup

A shortcut notation is available so the url link may also be provided automatically as a footnote

markup example:

```
about {
```

resulting output:

about SiSU [19] markup

11.5.3 Linking Images

markup example:

[19] <http://www.jus.uio.no/sisu/>

```
{tux.png 64x80 }image
% various url linked images
{tux.png 64x80 "a better way" }http://www.jus.uio.no/sisu/
{GnuDebianLinuxRubyBetterWay.png 100x101 "Way Better - with Gnu/Linux, Debian and Ruby" }http://www.jus.uio.no/sisu/
{
```

resulting output:

Gnu/Linux - a better way

[ruby_logo (png missing)] [20]

Way Better - with Gnu/Linux, Debian and Ruby

linked url footnote shortcut

```
{
text marker *ñame
```

note at a heading level the same is automatically achieved by providing names to headings 1, 2 and 3 i.e. 2~[name] and 3~[name] or in the case of auto-heading numbering, without further intervention.

11.6 Grouped Text

11.6.1 Tables

Tables may be prepared in two either of two forms

markup example:

```
table{ c3; 40; 30; 30;
```

[20] <http://www.ruby-lang.org/en/>

```
This is a table
this would become column two of row one
column three of row one is here
And here begins another row
column two of row two
column three of row two, and so on
}table
```

resulting output:

This is a table	this would become column two of row one	column three of row one is here
And here begins another row	column two of row two	column three of row two, and so on

a second form may be easier to work with in cases where there is not much information in each column

markup example: [21]

```
!_ Table 3.1: Contributors to Wikipedia, January 2001 - June 2005
{table~ h 24; 12; 12; 12; 12; 12; 12;}
                    |Jan. 2001|Jan. 2002|Jan. 2003|Jan. 2004|July 2004|June 2006
Contributors*              |     10|    472|   2,188|   9,653|  25,011|  48,721
Active contributors**      |      9|    212|    846|   3,228|   8,442|  16,945
Very active contributors***  |      0|     31|    190|    692|   1,639|   3,016
No. of English language articles|     25|  16,000|  101,000|  190,000|  320,000|  630,000
No. of articles, all languages  |     25|  19,000|  138,000|  490,000|  862,000|1,600,000
 Contributed at least ten times; * at least 5 times in last month; * more than 100 times in last month.
```

resulting output:

Table 3.1: Contributors to Wikipedia, January 2001 - June 2005

	Jan. 2001	Jan. 2002	Jan. 2003	Jan. 2004	July 2004	June 2006
Contributors*	10	472	2,188	9,653	25,011	48,721
Active contributors**	9	212	846	3,228	8,442	16,945
Very active contributors***	0	31	190	692	1,639	3,016
No. of English language articles	25	16,000	101,000	190,000	320,000	630,000
No. of articles, all languages	25	19,000	138,000	490,000	862,000	1,600,000

* Contributed at least ten times; ** at least 5 times in last month; *** more than 100 times in last month.

11.6.2 Poem

basic markup:

```
poem{
  Your poem here
}poem
```

Each verse in a poem is given a separate object number.

markup example:

```
poem{
          'Fury said to a
```

[21] Table from the Wealth of Networks by Yochai Benkler
<http://www.jus.uio.no/sisu/the_wealth_of_networks.yochai_benkler>

```
            mouse, That he
           met in the
          house,
        ”Let us
         both go to
          law:  I will
           prosecute
            YOU.  –Come,
             I’ll take no
              denial; We
             must have a
           trial:  For
         really this
       morning I’ve
      nothing
     to do.”
      Said the
       mouse to the
        cur, ”Such
         a trial,
          dear Sir,
               With
             no jury
           or judge,
          would be
         wasting
        our
         breath.”
         ”I’ll be
          judge, I’ll
           be jury,”
               Said
            cunning
             old Fury:
            ”I’ll
             try the
              whole
               cause,
                and
              condemn
             you
             to
             death.”’
}poem
```

resulting output:

```
            ‘Fury said to a
            mouse, That he
           met in the
          house,
        “Let us
```

both go to
law: I will
prosecute
YOU. –Come,
I'll take no
denial; We
must have a
trial: For
really this
morning I've
nothing
to do."
Said the
mouse to the
cur, "Such
a trial,
dear Sir,
With
no jury
or judge,
would be
wasting
our
breath."
"I'll be
judge, I'll
be jury,"
Said
cunning
old Fury:
"I'll
try the
whole
cause,
and
condemn
you
to
death."'

11.6.3 Group

basic markup:

```
group{
  Your grouped text here
}group
```

A group is treated as an object and given a single object number.

markup example:

```
group{
                ’Fury said to a
               mouse, That he
              met in the
             house,
           ”Let us
            both go to
             law:  I will
              prosecute
                YOU.  –Come,
                  I’ll take no
                   denial; We
                  must have a
              trial:  For
            really this
          morning I’ve
         nothing
        to do.”
          Said the
           mouse to the
            cur, ”Such
              a trial,
                dear Sir,
                     With
                  no jury
                 or judge,
                would be
               wasting
              our
               breath.”
                ”I’ll be
                  judge, I’ll
                    be jury,”
                       Said
                   cunning
                     old Fury:
                    ”I’ll
                     try the
                       whole
                        cause,
                          and
                       condemn
                        you
                       to
```

death."'
}group

resulting output:

'Fury said to a
mouse, That he
met in the
house,
"Let us
both go to
law: I will
prosecute
YOU. –Come,
I'll take no
denial; We
must have a
trial: For
really this
morning I've
nothing
to do."
Said the
mouse to the
cur, "Such
a trial,
dear Sir,
With
no jury
or judge,
would be
wasting
our
breath."
"I'll be
judge, I'll
be jury,"
Said
cunning
old Fury:

"I'll
try the
whole
cause,
and
condemn
you
to
death."'

11.6.4 Code

Code tags are used to escape regular sisu markup, and have been used extensively within this document to provide examples of **SiSU** markup. You cannot however use code tags to escape code tags. They are however used in the same way as group or poem tags.

A code-block is treated as an object and given a single object number. [an option to number each line of code may be considered at some later time]

use of code tags instead of poem compared, resulting output:

```
'Fury said to a
mouse, That he
met in the
house,
"Let us
both go to
law: I will
prosecute
YOU. –Come,
I'll take no
denial; We
must have a
trial: For
really this
morning I've
nothing
to do."
Said the
mouse to the
cur, "Such
a trial,
dear Sir,
With
no jury
or judge,
would be
wasting
our
```

breath.”
”I’ll be
judge, I’ll
be jury,”
Said
cunning
old Fury:
”I’ll
try the
whole
cause,
and
condemn
you
to
death.”’

12. Composite documents markup

It is possible to build a document by creating a master document that requires other documents. The documents required may be complete documents that could be generated independently, or they could be markup snippets, prepared so as to be easily available to be placed within another text. If the calling document is a master document (built from other documents), it should be named with the suffix **.ssm** Within this document you would provide information on the other documents that should be included within the text. These may be other documents that would be processed in a regular way, or markup bits prepared only for inclusion within a master document **.sst** regular markup file, or **.ssi** (insert/information) A secondary file of the composite document is built prior to processing with the same prefix and the suffix **._sst**

basic markup for importing a document into a master document

```
<< |filename1.sst|@|j
<< |filename2.ssi|@|j
```

The form described above should be relied on. Within the Vim editor it results in the text thus linked becoming hyperlinked to the document it is calling in which is convenient for editing. Alternative markup for importation of documents under consideration, and occasionally supported have been.

```
r{filename}
{filename.ssi}require
<< {filename.ssi}
% using textlink alternatives
|filename.ssi|@|jrequire
<< |filename.ssi|@|j
% using thlnk alternatives
<url:filename.ssi>require
<< <url:filename.ssi>
```

MARKUP SYNTAX HISTORY 392

13. Notes related to Files-types and Markup Syntax

0.38 is substantially current, depreciated 0.16 supported, though file names were changed at 0.37

0.52 (2007w14/6) declared document type identifier at start of text/document:

SiSU 0.52

or, backward compatible using the comment marker:

% **SiSU** 0.38

variations include '**SiSU** (text|master|insert) [version]' and 'sisu-[version]'

0.51 (2007w13/6) skins changed (simplified), markup unchanged

0.42 (2006w27/4) * (asterisk) type endnotes, used e.g. in relation to author

0.38 (2006w15/7) introduced new/alternative notation for headers, e.g. @title: (instead of 0~ title), and accompanying document structure markup, :A,:B,:C,1,2,3 (maps to previous 1,2,3,4,5,6)

0.37 (2006w09/7) introduced new file naming convention, .sst (text), .ssm (master), .ssi (insert), markup syntax unchanged

0.35 (2005w52/3) sisupod, zipped content file introduced

0.23 (2005w36/2) utf-8 for markup file

0.22 (2005w35/3) image dimensions may be omitted if rmagick is available to be relied upon

0.20.4 (2005w33/4) header 0~ links

0.16 (2005w25/2) substantial changes introduced to make markup cleaner, header 0~ title type, and headings [1-6]~ introduced, also percentage sign (%) at start of a text line as comment marker

14. SiSU filetypes

SiSU has plaintext and binary filetypes, and can process either type of document.

14.1 .sst .ssm .ssi marked up plain text

SiSU documents are prepared as plain-text (utf-8) files with **SiSU** markup. They may make reference to and contain images (for example), which are stored in the directory beneath them _sisu/image. **SiSU** plaintext markup files are of three types that may be distinguished by the file extension used: regular text .sst; master documents, composite documents that incorporate other text, which can be any regular text or text insert; and inserts the contents of which are like regular text except these are marked .ssi and are not processed.

SiSU processing can be done directly against a sisu documents; which may be located locally or on a remote server for which a url is provided.

SiSU source markup can be shared with the command:

sisu -s [filename]

14.1.1 sisu text - regular files (.sst)

The most common form of document in **SiSU** , see the section on **SiSU** markup.

<http://www.jus.uio.no/sisu/sisu_markup>

<http://www.jus.uio.no/sisu/sisu_manual>

14.1.2 sisu master files (.ssm)

Composite documents which incorporate other **SiSU** documents which may be either regular **SiSU** text .sst which may be generated independently, or inserts prepared solely for the purpose of being incorporated into one or more master documents.

The mechanism by which master files incorporate other documents is described as one of the headings under under **SiSU** markup in the **SiSU** manual.

Note: Master documents may be prepared in a similar way to regular documents, and processing will occur normally if a .sst file is renamed .ssm without requiring any other documents; the .ssm marker flags that the document may contain other documents.

Note: a secondary file of the composite document is built prior to processing with the same prefix and the suffix ._sst [22]

<http://www.jus.uio.no/sisu/sisu_markup>

<http://www.jus.uio.no/sisu/sisu_manual>

[22] .ssc (for composite) is under consideration but ._sst makes clear that this is not a regular file to be worked on, and thus less likely that people will have "accidents", working on a .ssc file that is overwritten by subsequent processing. It may be however that when the resulting file is shared .ssc is an appropriate suffix to use.

14.1.3 sisu insert files (.ssi)

Inserts are documents prepared solely for the purpose of being incorporated into one or more master documents. They resemble regular **SiSU** text files except they are ignored by the **SiSU** processor. Making a file a .ssi file is a quick and convenient way of flagging that it is not intended that the file should be processed on its own.

14.2 sisupod, zipped binary container (sisupod.zip, .ssp)

A sisupod is a zipped **SiSU** text file or set of **SiSU** text files and any associated images that they contain (this will be extended to include sound and multimedia-files)

SiSU plaintext files rely on a recognised directory structure to find contents such as images associated with documents, but all images for example for all documents contained in a directory are located in the sub-directory _sisu/image. Without the ability to create a sisupod it can be inconvenient to manually identify all other files associated with a document. A sisupod automatically bundles all associated files with the document that is turned into a pod.

The structure of the sisupod is such that it may for example contain a single document and its associated images; a master document and its associated documents and anything else; or the zipped contents of a whole directory of prepared **SiSU** documents.

The command to create a sisupod is:

```
sisu -S [filename]
```

Alternatively, make a pod of the contents of a whole directory:

```
sisu -S
```

SiSU processing can be done directly against a sisupod; which may be located locally or on a remote server for which a url is provided.

<http://www.jus.uio.no/sisu/sisu_commands>

<http://www.jus.uio.no/sisu/sisu_manual>

15. Experimental Alternative Input Representations

15.1 Alternative XML

SiSU offers alternative XML input representations of documents as a proof of concept, experimental feature. They are however not strictly maintained, and incomplete and should be handled with care.

convert from sst to simple xml representations (sax, dom and node):

sisu –to-sax [filename/wildcard] or sisu –to-sxs [filename/wildcard]

sisu –to-dom [filename/wildcard] or sisu –to-sxd [filename/wildcard]

sisu –to-node [filename/wildcard] or sisu –to-sxn [filename/wildcard]

convert to sst from any sisu xml representation (sax, dom and node):

sisu –from-xml2sst [filename/wildcard [.sxs.xml,.sxd.xml,sxn.xml]]

or the same:

sisu –from-sxml [filename/wildcard [.sxs.xml,.sxd.xml,sxn.xml]]

15.1.1 XML SAX representation

To convert from sst to simple xml (sax) representation:

sisu –to-sax [filename/wildcard] or sisu –to-sxs [filename/wildcard]

To convert from any sisu xml representation back to sst

sisu –from-xml2sst [filename/wildcard [.sxs.xml,.sxd.xml,sxn.xml]]

or the same:

sisu –from-sxml [filename/wildcard [.sxs.xml,.sxd.xml,sxn.xml]]

15.1.2 XML DOM representation

To convert from sst to simple xml (dom) representation:

sisu –to-dom [filename/wildcard] or sisu –to-sxd [filename/wildcard]

To convert from any sisu xml representation back to sst

sisu –from-xml2sst [filename/wildcard [.sxs.xml,.sxd.xml,sxn.xml]]

or the same:

sisu –from-sxml [filename/wildcard [.sxs.xml,.sxd.xml,sxn.xml]]

15.1.3 XML Node representation

To convert from sst to simple xml (node) representation:

sisu –to-node [filename/wildcard] or sisu –to-sxn [filename/wildcard]

To convert from any sisu xml representation back to sst

sisu –from-xml2sst [filename/wildcard [.sxs.xml,.sxd.xml,sxn.xml]]

or the same:

sisu –from-sxml [filename/wildcard [.sxs.xml,.sxd.xml,sxn.xml]]

16. Configuration

16.1 Determining the Current Configuration

Information on the current configuration of **SiSU** should be available with the help command:

sisu -v

which is an alias for:

sisu –help env

Either of these should be executed from within a directory that contains sisu markup source documents.

16.2 Configuration files (config.yml)

SiSU configration parameters are adjusted in the configuration file, which can be used to override the defaults set. This includes such things as which directory interim processing should be done in and where the generated output should be placed.

The **SiSU** configuration file is a yaml file, which means indentation is significant.

SiSU resource configuration is determined by looking at the following files if they exist:

./_sisu/sisurc.yml

~ /.sisu/sisurc.yml

/etc/sisu/sisurc.yml

The search is in the order listed, and the first one found is used.

In the absence of instructions in any of these it falls back to the internal program defaults.

Configuration determines the output and processing directories and the database access details.

If **SiSU** is installed a sample sisurc.yml may be found in /etc/sisu/sisurc.yml

17. Skins

Skins modify the default appearance of document output on a document, directory, or site wide basis. Skins are looked for in the following locations:

./_sisu/skin

~ /.sisu/skin

/etc/sisu/skin

Within the skin directory are the following the default sub-directories for document skins:

./skin/doc

./skin/dir

./skin/site

A skin is placed in the appropriate directory and the file named skin_[name].rb

The skin itself is a ruby file which modifies the default appearances set in the program.

17.1 Document Skin

Documents take on a document skin, if the header of the document specifies a skin to be used.

@skin: skin_united_nations

17.2 Directory Skin

A directory may be mapped on to a particular skin, so all documents within that directory take on a particular appearance. If a skin exists in the skin/dir with the same name as the document directory, it will automatically be used for each of the documents in that directory, (except where a document specifies the use of another skin, in the skin/doc directory).

A personal habit is to place all skins within the doc directory, and symbolic links as needed from the site, or dir directories as required.

17.3 Site Skin

A site skin, modifies the program default skin.

17.4 Sample Skins

With **SiSU** installed sample skins may be found in:

/etc/sisu/skin/doc and /usr/share/doc/sisu/sisu_markup_samples/dfsg/_sisu/skin/doc

(or equivalent directory) and if sisu-markup-samples is installed also under:

/usr/share/doc/sisu/sisu_markup_samples/non-free/_sisu/skin/doc

Samples of list.yml and promo.yml (which are used to create the right column list) may be found in:

/usr/share/doc/sisu/sisu_markup_samples/dfsg/_sisu/skin/yml (or equivalent directory)

18. CSS - Cascading Style Sheets (for html, XHTML and XML)

CSS files to modify the appearance of **SiSU** html, XHTML or XML may be placed in the configuration directory: ./_sisu/css; ~ /.sisu/css or; /etc/sisu/css and these will be copied to the output directories with the command sisu -CC.

The basic CSS file for html output is html.css, placing a file of that name in directory _sisu/css or equivalent will result in the default file of that name being overwritten.

HTML: html.css

XML DOM: dom.css

XML SAX: sax.css

XHTML: xhtml.css

The default homepage may use homepage.css or html.css

Under consideration is to permit the placement of a CSS file with a different name in directory _sisu/css directory or equivalent, and change the default CSS file that is looked for in a skin.[23]

[23] **SiSU** has worked this way in the past, though this was dropped as it was thought the complexity outweighed the flexibility, however, the balance was rather fine and this behaviour could be reinstated.

19. Organising Content

19.1 Directory Structure and Mapping

The output directory root can be set in the sisurc.yml file. Under the root, subdirectories are made for each directory in which a document set resides. If you have a directory named poems or conventions, that directory will be created under the output directory root and the output for all documents contained in the directory of a particular name will be generated to subdirectories beneath that directory (poem or conventions). A document will be placed in a subdirectory of the same name as the document with the filetype identifier stripped (.sst .ssm)

The last part of a directory path, representing the sub-directory in which a document set resides, is the directory name that will be used for the output directory. This has implications for the organisation of document collections as it could make sense to place documents of a particular subject, or type within a directory identifying them. This grouping as suggested could be by subject (sales_law, english_literature); or just as conveniently by some other classification (X University). The mapping means it is also possible to place in the same output directory documents that are for organisational purposes kept separately, for example documents on a given subject of two different institutions may be kept in two different directories of the same name, under a directory named after each institution, and these would be output to the same output directory. Skins could be associated with each institution on a directory basis and resulting documents will take on the appropriate different appearance.

19.2 Organising Content

20. Homepages

SiSU is about the ability to auto-generate documents. Home pages are regarded as custom built items, and are not created by **SiSU** . More accurately, **SiSU** has a default home page, which will not be appropriate for use with other sites, and the means to provide your own home page instead in one of two ways as part of a site's configuration, these being:

1. through placing your home page and other custom built documents in the subdirectory _sisu/home/ (this probably being the easier and more convenient option)

2. through providing what you want as the home page in a skin,

Document sets are contained in directories, usually organised by site or subject. Each directory can/should have its own homepage. See the section on directory structure and organisation of content.

20.1 Home page and other custom built pages in a sub-directory

Custom built pages, including the home page index.html may be placed within the configuration directory _sisu/home/ in any of the locations that is searched for the configuration directory, namely ./_sisu; ~ /_sisu; /etc/sisu From there they are copied to the root of the output directory with the command:

sisu -CC

20.2 Home page within a skin

Skins are described in a separate section, but basically are a file written in the programming language **Ruby** that may be provided to change the defaults that are provided with sisu with respect to individual documents, a directories contents or for a site.

If you wish to provide a homepage within a skin the skin should be in the directory _sisu/skin/dir and have the name of the directory for which it is to become the home page. Documents in the directory commercial_law would have the homepage modified in skin_commercial law.rb; or the directory poems in skin_poems.rb

```
class Home
  def homepage
    # place the html content of your homepage here, this will become index.html
    <<HOME <html>
<head></head>
<doc>
<p>this is my new homepage.</p>
</doc>
</html>
HOME
  end
end
```

21. Markup and Output Examples 542

21.1 Markup examples 543

Current markup examples and document output samples are provided at <http://www.jus.uio.no/sisu/SiSU/2.html> 544

Some markup with syntax highlighting may be found under <http://www.jus.uio.no/sisu/sample/syntax> but is not as up to date. 545

For some documents hardly any markup at all is required at all, other than a header, and an indication that the levels to be taken into account by the program in generating its output are. 546

21.2 A few book (and other) examples

Aukio, by Leena Krohn [24]

"The Wealth of Networks", Yochai Benkler

"The Wealth of Networks", Yochai Benkler

- document manifest [25]
- html, segmented text
- html, scroll, document in one
- pdf, landscape
- pdf, portrait
- open document
- xhtml scroll

[24] Reproduced with the kind permission of author and artist Leena Krohn, <http://www.kaapeli.fi/krohn> "Aukio" is from the work *"Sphinx or Robot"* <http://www.jus.uio.no/sisu/sphinx_or_robot.leena_krohn.1996> which is included as a book example in this section, together with another of the author's works, *"Tainaron"* <http://www.jus.uio.no/sisu/tainaron.leena_krohn.1998>

[25] <http://www.jus.uio.no/sisu/sisu_manual/the_wealth_of_networks.yochai_benkler/sisu_manifest.html>

xml, sax 558
xml, dom 559
plain text utf-8 560
concordance 561
dcc, document content certificate (digests) 562
markup source text 563
zipped markup source pod 564

***"Free Culture"*, Lawrence Lessig** 565

"Free Culture", Lawrence Lessig 566
document manifest [26] 567
html, segmented text 568
html, scroll, document in one 569
pdf, landscape 570
pdf, portrait 571
open document 572
xhtml scroll 573
xml, sax 574
xml, dom 575
plain text utf-8 576
concordance 577
dcc, document content certificate (digests) 578
markup source text 579
zipped markup source pod 580

***"Free as in Freedom: Richard Stallman's Crusade for Free Software"*, by Sam Williams** 581

"Free as in Freedom: Richard Stallman's Crusade for Free Software", by Sam Williams 582
document manifest [27] 583
html, segmented text 584
html, scroll, document in one 585
pdf, landscape 586

pdf, portrait 587
open document 588
xhtml scroll 589
xml, sax 590
xml, dom 591
plain text utf-8 592
concordance 593
dcc, document content certificate (digests) 594
markup source text 595
zipped markup source pod 596

***"Free For All: How Linux and the Free Software Movement Undercut the High Tech Titans"*, by Peter Wayner** 597

"Free For All: How Linux and the Free Software Movement Undercut the High Tech Titans", by Peter Wayner 598
document manifest [28] 599
html, segmented text 600
html, scroll, document in one 601
pdf, landscape 602
pdf, portrait 603
open document 604
xhtml scroll 605
xml, sax 606
xml, dom 607
plain text utf-8 608
concordance 609
dcc, document content certificate (digests) 610
markup source text 611
zipped markup source pod 612

***"The Cathedral and the Bazaar"*, by Eric S. Raymond** 613

"The Cathedral and the Bazaar", by Eric S. Raymond 614

- document manifest [29] 615
- html, segmented text 616
- html, scroll, document in one 617
- pdf, landscape 618
- pdf, portrait 619
- open document 620
- xhtml scroll 621
- xml, sax 622
- xml, dom 623
- plain text utf-8 624
- concordance 625
- dcc, document content certificate (digests) 626
- markup source text 627
- zipped markup source pod 628

***"Accelerando"*, Charles Stross** 629

"Accelerando", Charles Stross 630

- document manifest [30] 631
- html, segmented text 632
- html, scroll, document in one 633
- pdf, landscape 634
- pdf, portrait 635
- open document 636
- xhtml scroll 637
- xml, sax 638
- xml, dom 639
- plain text utf-8 640
- concordance 641

[26] <http://www.jus.uio.no/sisu/sisu_manual/free_culture.lawrence_lessig/sisu_manifest.html>
[27] <http://www.jus.uio.no/sisu/sisu_manual/free_as_in_freedom.richard_stallman_crusade_for_free_software.sam_williams/sisu_manifest.html>
[28] <http://www.jus.uio.no/sisu/sisu_manual/free_for_all.peter_wayner/sisu_manifest.html>
[29] <http://www.jus.uio.no/sisu/sisu_manual/the_cathedral_and_the_bazaar.eric_s_raymond/sisu_manifest.html>
[30] <http://www.jus.uio.no/sisu/sisu_manual/accelerando.charles_stross/sisu_manifest.html>

dcc, document content certificate (digests)

markup source text

zipped markup source pod

"Tainaron", Leena Krohn

"Tainaron", Leena Krohn

document manifest [31]

html, segmented text

html, scroll, document in one

pdf, landscape

pdf, portrait

open document

xhtml scroll

xml, sax

xml, dom

plain text utf-8

concordance

dcc, document content certificate (digests)

markup source text

zipped markup source pod

"Sphinx or Robot", Leena Krohn

Sphinx or Robot by Leena Krohn

"Sphinx or Robot", Leena Krohn

document manifest [32]

html, segmented text

html, scroll, document in one

pdf, landscape

pdf, portrait 668

open document 669

xhtml scroll 670

xml, sax 671

xml, dom 672

plain text utf-8 673

concordance 674

dcc, document content certificate (digests) 675

markup source text 676

zipped markup source pod 677

"War and Peace", Leo Tolstoy, PG Etext 2600 678

"War and Peace", Leo Tolstoy [33] 679

document manifest [34] 680

html, segmented text 681

html, scroll, document in one 682

pdf, landscape 683

pdf, portrait 684

open document 685

[31] <http://www.jus.uio.no/sisu/sisu_manual/tainaron.leena_krohn.1998/sisu_manifest.html>

[32] <http://www.jus.uio.no/sisu/sisu_manual/sphinx_or_robot.leena_krohn.1996/sisu_manifest.html>

[33] <http://www.jus.uio.no/sisu/war_and_peace.leo_tolstoy/toc.html>
The ascii text was taken from Project Gutenberg. The markup transforms required are trivial. Of interest, in this instance I am saved by having alternative syntaxes/(structural modes) for marking up endnotes... as it was possible to do a simple search and replace to make the Project Gutenberg ascii presentation suitable for **SiSU** , using the older endnote markup style. This example instructs the program to use regular expressions, in this example the words: none; none; BOOK|FIRST|SECOND; CHAPTER; occurring at the beginning of a line, to identify what should be treated as different levels of heading in a document (and used to make the table of contents). Note that there was very little markup required after the document headers and Project Gutenberg legal notices. As I presume the legal notices are similar in Project Gutenberg documents, (and I could not bear to think of preparing the same legal notices twice), I moved those to the "skin" for the Project, and these are now represented in the markup by <:insert1> and <:insert2> and the legal notices are available for similar insertion into the next Project Gutenberg text prepared for **SiSU** , should there be one.
I did a stylesheet/skin for the Gutenberg Project, ... I may have to remove. The markup transforms required are trivial. Of interest, in this instance I am saved by having alternative syntaxes/(structural modes) for marking up endnotes... as it is possible to do a simple search and replace to make Project Gutenberg ascii presentations suitable for **SiSU** using the older endnote markup style. There is very little markup required after the document headers and Project Gutenberg legal notices. As I presume the legal notices are similar in Project Gutenberg documents, (and I could not bear to think of preparing the same legal notices twice), I moved those to the "skin" for the Project, and these are now represented in the markup by the <:insert1> and <:insert2> markers and the legal notices are available for similar insertion into the next Project Gutenberg text prepared for **SiSU** , should there be one.

[34] <http://www.jus.uio.no/sisu/sisu_manual/war_and_peace.leo_tolstoy/sisu_manifest.html>

xhtml scroll 686
xml, sax 687
xml, dom 688
plain text utf-8 689
concordance 690
dcc, document content certificate (digests) 691
markup source text 692
zipped markup source pod 693

***"Don Quixote"*, Miguel de Cervantes [Saavedra], translated by John Ormsby, PG Etext 996** 694

"Don Quixote", Miguel de Cervantes [Saavedra] 695
document manifest [35] 696
html, segmented text 697
html, scroll, document in one 698
pdf, landscape 699
pdf, portrait 700
open document 701
xhtml scroll 702
xml, sax 703
xml, dom 704
plain text utf-8 705
concordance 706
dcc, document content certificate (digests) 707
markup source text 708
zipped markup source pod 709

***"Gulliver's Travels"*, Jonathan Swift, transcribed from the 1892 George Bell and Sons edition by David Price, PG Etext 829** 710

"Gulliver's Travels", Jonathan Swift 711
document manifest [36] 712
html, segmented text 713

html, scroll, document in one 714

pdf, landscape 715

pdf, portrait 716

open document 717

xhtml scroll 718

xml, sax 719

xml, dom 720

plain text utf-8 721

concordance 722

dcc, document content certificate (digests) 723

markup source text 724

zipped markup source pod 725

***"Alice's Adventures in Wonderland"*, Lewis Carroll, PG Etext 11** 726

"Alice's Adventures in Wonderland", Lewis Carroll 727

document manifest [37] 728

html, segmented text 729

html, scroll, document in one 730

pdf, landscape 731

pdf, portrait 732

open document 733

xhtml scroll 734

xml, sax 735

xml, dom 736

plain text utf-8 737

concordance 738

dcc, document content certificate (digests) 739

markup source text 740

zipped markup source pod 741

***"Through The Looking-Glass"*, Lewis Carroll, PG Etext 12** 742

"Through The Looking-Glass", Lewis Carroll 743

[35] <http://www.jus.uio.no/sisu/sisu_manual/don_quixote.miguel_de_cervantes/sisu_manifest.html>

[36] <http://www.jus.uio.no/sisu/sisu_manual/gullivers_travels.jonathan_swift/sisu_manifest.html>

[37] <http://www.jus.uio.no/sisu/sisu_manual/alices_adventures_in_wonderland.lewis_carroll/sisu_manifest.html>

document manifest [38]

html, segmented text

html, scroll, document in one

pdf, landscape

pdf, portrait

open document

xhtml scroll

xml, sax

xml, dom

plain text utf-8

concordance

dcc, document content certificate (digests)

markup source text

zipped markup source pod

"Alice's Adventures in Wonderland" and *"Through The Looking-Glass"*, Lewis Carroll, PG Etexts 11 and 12

"Alice's Adventures in Wonderland" and "Through The Looking-Glass", Lewis Carroll

document manifest [39]

html, segmented text

html, scroll, document in one

pdf, landscape

pdf, portrait

open document

xhtml scroll

xml, sax

xml, dom

plain text utf-8

concordance

dcc, document content certificate (digests)

markup source text

zipped markup source pod

[38] <http://www.jus.uio.no/sisu/sisu_manual/through_the_looking_glass.lewis_carroll/sisu_manifest.html>

[39] <http://www.jus.uio.no/sisu/sisu_manual/alices_adventures_in_wonderland_and_through_the_looking_glass.lewis_carroll/sisu_manifest.html>

***"Gnu Public License 2"*, (GPL 2) Free Software Foundation** 774

"Gnu Public License 2", (GPL 2) Free Software Foundation 775

document manifest [40] 776

html, segmented text 777

html, scroll, document in one 778

pdf, landscape 779

pdf, portrait 780

open document 781

xhtml scroll 782

xml, sax 783

xml, dom 784

plain text utf-8 785

concordance 786

dcc, document content certificate (digests) 787

markup source text 788

zipped markup source pod 789

***"Gnu Public License v3 - Third discussion draft"*, (GPLv3) Free Software Foundation** 790

"Gnu Public License 3 - Third discussion draft", (GPL v3 draft3) Free Software Foundation 791

document manifest [41] 792

html, segmented text 793

html, scroll, document in one 794

pdf, landscape 795

pdf, portrait 796

open document 797

xhtml scroll 798

xml, sax 799

xml, dom 800

plain text utf-8 801

concordance 802

dcc, document content certificate (digests) 803
markup source text 804
zipped markup source pod 805

"Debian Social Contract" 806

"Debian Social Contract" 807
document manifest [42] 808
html, segmented text 809
html, scroll, document in one 810
pdf, landscape 811
pdf, portrait 812
open document 813
xhtml scroll 814
xml, sax 815
xml, dom 816
plain text utf-8 817
concordance 818
dcc, document content certificate (digests) 819
markup source text 820
zipped markup source pod 821

"Debian Constitution v1.3"**, (simple/default markup)** 822

"Debian Constitution v1.3" 823
document manifest [43] 824
html, segmented text 825
html, scroll, document in one 826
pdf, landscape 827
pdf, portrait 828
open document 829
xhtml scroll 830

xml, sax

xml, dom

plain text utf-8

concordance

dcc, document content certificate (digests)

markup source text

zipped markup source pod

***"Debian Constitution v1.3"*, (markup adjusted for output to more closely match the original)**

"Debian Constitution v1.3", (markup adjusted for output to more closely match the original)

document manifest [44]

html, segmented text

html, scroll, document in one

pdf, landscape

pdf, portrait

open document

xhtml scroll

xml, sax

xml, dom

plain text utf-8

concordance

dcc, document content certificate (digests)

markup source text

zipped markup source pod

***"Debian Constitution v1.2"*, (simple/default markup)**

"Debian Constitution v1.2 (more translations)"

document manifest [45]

[40] <http://www.jus.uio.no/sisu/sisu_manual/gpl2.fsf/sisu_manifest.html>

[41] <http://www.jus.uio.no/sisu/sisu_manual/gpl3_draft3.fsf/sisu_manifest.html>

[42] <http://www.jus.uio.no/sisu/sisu_manual/debian_social_contract_v1.1/sisu_manifest.html>

[43] <http://www.jus.uio.no/sisu/sisu_manual/debian_constitution_v1.3/sisu_manifest.html>

[44] <http://www.jus.uio.no/sisu/sisu_manual/debian_constitution_v1.3.adjusted/sisu_manifest.html>

[45] <http://www.jus.uio.no/sisu/sisu_manual/debian_constitution_v1.2/sisu_manifest.html>

html, segmented text 857
html, scroll, document in one 858
pdf, landscape 859
pdf, portrait 860
open document 861
xhtml scroll 862
xml, sax 863
xml, dom 864
plain text utf-8 865
concordance 866
dcc, document content certificate (digests) 867
markup source text 868
zipped markup source pod 869

"*Debian Constitution v1.2*", (markup adjusted for output to more closely match the original) 870

"Debian Constitution (more translations)", (markup adjusted for output to more closely match the original) 871
document manifest [46] 872
html, segmented text 873
html, scroll, document in one 874
pdf, landscape 875
pdf, portrait 876
open document 877
xhtml scroll 878
xml, sax 879
xml, dom 880
plain text utf-8 881
concordance 882
dcc, document content certificate (digests) 883
markup source text 884
zipped markup source pod 885

***"A Uniform Sales Terminology"*, Vikki Rogers and Albert Kritzer** 886

"A Uniform Sales Terminology", Vikki Rogers and Albert Kritzer 887

document manifest [47] 888

html, segmented text 889

html, scroll, document in one 890

pdf, landscape 891

pdf, portrait 892

open document 893

xhtml scroll 894

xml, sax 895

xml, dom 896

plain text utf-8 897

concordance 898

dcc, document content certificate (digests) 899

markup source text 900

zipped markup source pod 901

***"The Autonomous Contract" 1997* - markup sample** 902

"The Autonomous Contract" 1997 - markup sample 903

document manifest [48] 904

html, segmented text 905

html, scroll, document in one 906

pdf, landscape 907

pdf, portrait 908

open document 909

xhtml scroll 910

xml, sax 911

xml, dom 912

plain text utf-8 913

concordance 914

dcc, document content certificate (digests) 915
markup source text 916
zipped markup source pod 917

***"The Autonomous Contract Revisited"* - markup sample** 918

"The Autonomous Contract Revisited" - markup sample [49] 919
document manifest [50] 920
html, segmented text 921
html, scroll, document in one 922
pdf, landscape 923
pdf, portrait 924
open document 925
xhtml scroll 926
xml, sax 927
xml, dom 928
plain text utf-8 929
concordance 930
dcc, document content certificate (digests) 931
markup source text 932
zipped markup source pod 933

"United Nations Convention on Contracts for the International Sale of Goods" 934

"United Nations Convention on Contracts for the International Sale of Goods" [51] 935

document manifest [52] 936

html, segmented text 937

html, scroll, document in one 938

pdf, landscape 939

pdf, portrait 940

open document 941

xhtml scroll 942

xml, sax 943

xml, dom 944

plain text utf-8 945

concordance 946

dcc, document content certificate (digests) 947

markup source text 948

zipped markup source pod 949

---

[46] <http://www.jus.uio.no/sisu/sisu_manual/debian_constitution_v1.2.adjusted/sisu_manifest.html>

[47] <http://www.jus.uio.no/sisu/sisu_manual/a_uniform_international_sales_terminology.vikki_rogers.and.albert_kritzer/sisu_manifest.html>

[48] <http://www.jus.uio.no/sisu/sisu_manual/the_autonomous_contract.amissah.19970710/sisu_manifest.html>

[49] <http://www.jus.uio.no/sisu/autonomy_markup0/toc.html>
alternative markup variations revolving around endnotes
(i) as above, markup with embedded endnotes, and header list of words/phrases to emphasise
<http://www.jus.uio.no/sisu/sample/syntax/autonomy_markup0.sst.html>
<http://www.jus.uio.no/sisu/sample/markup/autonomy_markup0.sst>
(ii) Again markup with embedded endnotes, but font faces changed within paragraphs rather than in header as in i
<http://www.jus.uio.no/sisu/sample/syntax/autonomy_markup1.sst.html>
<http://www.jus.uio.no/sisu/sample/markup/autonomy_markup1.sst>
(iii) Markup with endnote placemarks within paragraphs, the endnotes following the paragraph that contains them
<http://www.jus.uio.no/sisu/sample/syntax/autonomy_markup2.sst.html>
<http://www.jus.uio.no/sisu/sample/markup/autonomy_markup2.sst>
(iv) Another alternative is to place the marked up endnotes sequentially and at the end of the text. This also works. The paragraph variant iii is perhaps easier to visually check should there be missing endnotes; but this variant iv may better suit the conversion of alternatively pre-prepared documents.

[50] <http://www.jus.uio.no/sisu/sisu_manual/autonomy_markup0/sisu_manifest.html>

[51] <http://www.jus.uio.no/sisu/un_contracts_international_sale_of_goods_convention_1980/toc.html>
This example instructs the program to use regular expressions, in this example the words: Part, Chapter, Section, Article occurring at the beginning of a line, to identify what should be treated as different levels of heading in a document (and used to make the table of contents).
This example instructs the program to use regular expressions, in this example the words: Part, Chapter, Section, Article occurring at the beginning of a line, to identify what should be treated as different levels of heading in a document (and used to make the table of contents).

[52] <http://www.jus.uio.no/sisu/sisu_manual/un_contracts_international_sale_of_goods_convention_1980/sisu_manifest.html>

***PECL* the "Principles of European Contract Law"** 950

"Principles of European Contract Law" 951

- document manifest [53] 952
- html, segmented text 953
- html, scroll, document in one 954
- pdf, landscape 955
- pdf, portrait 956
- open document 957
- xhtml scroll 958
- xml, sax 959
- xml, dom 960
- plain text utf-8 961
- concordance 962
- dcc, document content certificate (digests) 963
- markup source text 964
- zipped markup source pod 965

21.3 SQL - PostgreSQL, SQLite 966

A Sample search form is available at <http://search.sisudoc.org> 967

A few canned searches, showing object numbers. Search for: 968

- English documents matching Linux OR Debian 969
- GPL OR Richard Stallman 970
- invention OR innovation in English language 971
- copyright in English language documents 972

Note that the searches done in this form are case sensitive. 973

Expand those same searches, showing the matching text in each document: 974

- English documents matching Linux OR Debian 975
- GPL OR Richard Stallman 976
- invention OR innovation in English language 977
- copyright in English language documents 978

Note you may set results either for documents matched and object number locations within each matched document meeting the search criteria; or display the names of the documents matched along with the objects (paragraphs) that meet the search criteria.[54]

21.4 Lex Mercatoria as an example

There is quite a bit to peruse if you explore the site Lex Mercatoria:

<http://www.lexmercatoria.org/> [55]

or perhaps:

<http://lexmercatoria.org/treaties.and.organisations/lm.chronological> [56]

21.5 For good measure the markup for a document with lots of (simple) tables

SiSU is not optimised for table making, but does handle simple tables.

- SiSU marked up file with tables [57]
- Output of table file example [58]

21.6 And a link to the output of a reported case

<http://www.jus.uio.no/lm/england.fothergill.v.monarch.airlines.hl.1980/toc.html>

[53] <http://www.jus.uio.no/sisu/sisu_manual/eu_contract_principles_parts_1_to_3_2002/sisu_manifest.html>

[54] of this feature when demonstrated to an IBM software innovations evaluator in 2004 he said to paraphrase: this could be of interest to us. We have large document management systems, you can search hundreds of thousands of documents and we can tell you which documents meet your search criteria, but there is no way we can tell you without opening each document where within each your matches are found.

[55] <http://www.jus.uio.no/lm/index>

[56] <http://www.jus.uio.no/lm/treaties.and.organisations/lm.chronological>

[57] <http://www.jus.uio.no/sisu/sample/syntax/un_conventions_membership_status.sst.html>
<http://www.jus.uio.no/sisu/sample/markup/un_conventions_membership_status.sst>

[58] <http://www.jus.uio.no/lm/un_conventions_membership_status/toc.html>

22. A Checklist of Output Features

This table gives an indication of the features that are available for various forms of output of **SiSU** . [59]

feature	txt	ltx/pdf	HTML	XHTML	XML/s	XML/d	ODF	SQLite	pgSQL
headings	*	*	*	*	*	*	*	*	*
footnotes	*	*	*	*	*	*	*	*	*
bold, underscore, italics	.	*	*	*	*	*	*	*	*
strikethrough	.	*	*	*	*	*	*		
superscript, subscript	.	*	*	*	*	*	*		
extended ascii set (utf-8)	*	*	*	*	*	*	*		*
indents	*	*	*	*	*	*	*		
bullets	.	*	*	*	*	*	.		
groups									
* tables		*	*	.	.	.	.	.	.
* poem	*	*	*	.	.	.	*	.	.
* code	*	*	*	.	.	.	*	.	.
url	*	*	*	*	*	*	*	.	.
links	*	*	*	*	*	*	*	.	.
images	-	*	*	T	T	T	*	T	T
image caption	-	*	*						
table of contents		*	*	*	*	*	.		
page header/footer?	-	*	*	*	*	*	t		
line break	*	*	*	*	*	*	*		
page break		*					*		
segments			*						
skins	*	*	*	*	*	*			
ocn	.	*	*	*	*	*	-?	*	*
auto-heading numbers	*	*	*	*	*	*	*	*	*
minor list numbering	*	*	*	*	*	*	*	*	*
special characters	.	.	.						

Done

* yes/done

. partial

- not available/appropriate

Not Done

T task todo

t lesser task/todo

not done

[59] updated for sisu-0.36.6 on 2006-01-23

23. SiSU Search - Introduction

SiSU output can easily and conveniently be indexed by a number of standalone indexing tools, such as Lucene, Hyperestraier.

Because the document structure of sites created is clearly defined, and the text object citation system is available hypothetically at least, for all forms of output, it is possible to search the sql database, and either read results from that database, or just as simply map the results to the html output, which has richer text markup.

In addition to this **SiSU** has the ability to populate a relational sql type database with documents at an object level, with objects numbers that are shared across different output types, which make them searchable with that degree of granularity. Basically, your match criteria is met by these documents and at these locations within each document, which can be viewed within the database directly or in various output formats.

24. SQL

24.1 populating SQL type databases

SiSU feeds sisu markupd documents into sql type databases PostgreSQL[60] and/or SQLite[61] database together with information related to document structure.

This is one of the more interesting output forms, as all the structural data of the documents are retained (though can be ignored by the user of the database should they so choose). All site texts/documents are (currently) streamed to four tables:

- one containing semantic (and other) headers, including, title, author, subject, (the Dublin Core...);
- another the substantive texts by individual "paragraph" (or object) - along with structural information, each paragraph being identifiable by its paragraph number (if it has one which almost all of them do), and the substantive text of each paragraph quite naturally being searchable (both in formatted and clean text versions for searching); and
- a third containing endnotes cross-referenced back to the paragraph from which they are referenced (both in formatted and clean text versions for searching).
- a fourth table with a one to one relation with the headers table contains full text versions of output, eg. pdf, html, xml, and ascii.

There is of course the possibility to add further structures.

At this level **SiSU** loads a relational database with documents chunked into objects, their smallest logical structurally constituent parts, as text objects, with their object citation number and all other structural information needed to construct the document. Text is stored (at this text object level) with and without elementary markup tagging, the stripped version being so as to facilitate ease of searching.

Being able to search a relational database at an object level with the **SiSU** citation system is an effective way of locating content generated by **SiSU** . As individual text objects of a document stored (and indexed) together with object numbers, and all versions of the document have the same numbering, complex searches can be tailored to return just the locations of the search results relevant for all available output formats, with live links to the precise locations in the database or in html/xml documents; or, the structural information provided makes it possible to search the full contents of the database and have headings in which search content appears, or to search only headings etc. (as the Dublin Core is incorporated it is easy to make use of that as well).

[60] <http://www.postgresql.org/>
<http://advocacy.postgresql.org/>
<http://en.wikipedia.org/wiki/Postgresql>

[61] <http://www.hwaci.com/sw/sqlite/>
<http://en.wikipedia.org/wiki/Sqlite>

25. Postgresql

25.1 Name

SiSU - Structured information, Serialized Units - a document publishing system, postgresql dependency package

25.2 Description

Information related to using postgresql with sisu (and related to the sisu_postgresql dependency package, which is a dummy package to install dependencies needed for **SiSU** to populate a postgresql database, this being part of **SiSU** - man sisu).

25.3 Synopsis

sisu -D [instruction] [filename/wildcard if required]

sisu -D –pg –[instruction] [filename/wildcard if required]

25.4 Commands

Mappings to two databases are provided by default, postgresql and sqlite, the same commands are used within sisu to construct and populate databases however -d (lowercase) denotes sqlite and -D (uppercase) denotes postgresql, alternatively –sqlite or –pgsql may be used

-D or –pgsql may be used interchangeably.

25.4.1 create and destroy database

–pgsql –createall
initial step, creates required relations (tables, indexes) in existing (postgresql) database (a database should be created manually and given the same name as working directory, as requested) (rb.dbi)

sisu -D –createdb
creates database where no database existed before

sisu -D –create
creates database tables where no database tables existed before

sisu -D –Dropall
destroys database (including all its content)! kills data and drops tables, indexes and database associated with a given directory (and directories of the same name).

sisu -D –recreate
destroys existing database and builds a new empty database structure

25.4.2 import and remove documents

sisu -D –import -v [filename/wildcard]
populates database with the contents of the file. Imports documents(s) specified to a postgresql database (at an object level).

sisu -D –update -v [filename/wildcard]
updates file contents in database

sisu -D –remove -v [filename/wildcard]
removes specified document from postgresql database.

26. Sqlite

26.1 Name

SiSU - Structured information, Serialized Units - a document publishing system.

26.2 Description

Information related to using sqlite with sisu (and related to the sisu_sqlite dependency package, which is a dummy package to install dependencies needed for **SiSU** to populate an sqlite database, this being part of **SiSU** - man sisu).

26.3 Synopsis

sisu -d [instruction] [filename/wildcard if required]

sisu -d –(sqlite|pg) –[instruction] [filename/wildcard if required]

26.4 Commands

Mappings to two databases are provided by default, postgresql and sqlite, the same commands are used within sisu to construct and populate databases however -d (lowercase) denotes sqlite and -D (uppercase) denotes postgresql, alternatively –sqlite or –pgsql may be used

-d or –sqlite may be used interchangeably.

26.4.1 create and destroy database

–sqlite –createall
initial step, creates required relations (tables, indexes) in existing (sqlite) database (a database should be created manually and given the same name as working directory, as requested) (rb.dbi)

sisu -d –createdb
creates database where no database existed before

sisu -d –create
creates database tables where no database tables existed before

sisu -d –dropall
destroys database (including all its content)! kills data and drops tables, indexes and database associated with a given directory (and directories of the same name).

sisu -d –recreate
destroys existing database and builds a new empty database structure

26.4.2 import and remove documents

sisu -d –import -v [filename/wildcard]
populates database with the contents of the file. Imports documents(s) specified to an sqlite database (at an object level).

sisu -d –update -v [filename/wildcard]
updates file contents in database

sisu -d –remove -v [filename/wildcard]
removes specified document from sqlite database.

27. Introduction

27.1 Search - database frontend sample, utilising database and SiSU features, including object citation numbering (backend currently PostgreSQL)

Sample search frontend [62] A small database and sample query front-end (search from) that makes use of the citation system, object citation numbering to demonstrates functionality.[63]

SiSU can provide information on which documents are matched and at what locations within each document the matches are found. These results are relevant across all outputs using object citation numbering, which includes html, XML, LaTeX, PDF and indeed the SQL database. You can then refer to one of the other outputs or in the SQL database expand the text within the matched objects (paragraphs) in the documents matched.

Note you may set results either for documents matched and object number locations within each matched document meeting the search criteria; or display the names of the documents matched along with the objects (paragraphs) that meet the search criteria.[64]

sisu -F –webserv-webrick
builds a cgi web search frontend for the database created

The following is feedback on the setup on a machine provided by the help command:

sisu –help sql

```
Postgresql
  user:          ralph
  current db set:  SiSU_sisu
  port:          5432
  dbi connect:     DBI:Pg:database=SiSU_sisu;port=5432
sqlite
  current db set:  /home/ralph/sisu_www/sisu/sisu_sqlite.db
  dbi connect      DBI:SQLite:/home/ralph/sisu_www/sisu/sisu_sqlite.db
```

Note on databases built

By default, [unless otherwise specified] databases are built on a directory basis, from collections of documents within that directory. The name of the directory you choose to work from is used as the database name, i.e. if you are working in a directory called /home/ralph/ebook the database SiSU_ebook is used. [otherwise a manual mapping for the collection is necessary]

27.2 Search Form

sisu -F

[62] <http://search.sisudoc.org>

[63] (which could be extended further with current back-end). As regards scaling of the database, it is as scalable as the database (here Postgresql) and hardware allow.

[64] of this feature when demonstrated to an IBM software innovations evaluator in 2004 he said to paraphrase: this could be of interest to us. We have large document management systems, you can search hundreds of thousands of documents and we can tell you which documents meet your search criteria, but there is no way we can tell you without opening each document where within each your matches are found.

generates a sample search form, which must be copied to the web-server cgi directory

sisu -F –webserv-webrick
generates a sample search form for use with the webrick server, which must be copied to the web-server cgi directory

sisu -Fv
as above, and provides some information on setting up hyperestraier

sisu -W
starts the webrick server which should be available wherever sisu is properly installed

The generated search form must be copied manually to the webserver directory as instructed

28. Hyperestraier

See the documentation for hyperestraier:

＜http://hyperestraier.sourceforge.net/＞

/usr/share/doc/hyperestraier/index.html

man estcmd

on sisu_hyperestraier:

man sisu_hyperestraier

/usr/share/doc/sisu/sisu_markup/sisu_hyperestraier/index.html

NOTE: the examples that follow assume that sisu output is placed in the directory /home/ralph/sisu_www

(A) to generate the index within the webserver directory to be indexed:

estcmd gather -sd [index name] [directory path to index]

the following are examples that will need to be tailored according to your needs:

cd /home/ralph/sisu_www

estcmd gather -sd casket /home/ralph/sisu_www

you may use the 'find' command together with 'egrep' to limit indexing to particular document collection directories within the web server directory:

find /home/ralph/sisu_www -type f | egrep '/home/ralph/sisu_www/sisu/.+?.html$' |estcmd gather -sd casket -

Check which directories in the webserver/output directory (~ /sisu_www or elsewhere depending on configuration) you wish to include in the search index.

As sisu duplicates output in multiple file formats, it it is probably preferable to limit the estraier index to html output, and as it may also be desirable to exclude files 'plain.txt', 'toc.html' and 'concordance.html', as these duplicate information held in other html output e.g.

find /home/ralph/sisu_www -type f | egrep '/sisu_www/(sisu|bookmarks)/.+?.html$' | egrep -v '(doc|concordance).html$' |estcmd gather -sd casket -

from your current document preparation/markup directory, you would construct a rune along the following lines:

find /home/ralph/sisu_www -type f | egrep '/home/ralph/sisu_www/([specify first directory for inclusion]|[specify second directory for inclusion]|[another directory for inclusion? ...])/.+?.html$' | egrep -v '(doc|concordance).html$' |estcmd gather -sd /home/ralph/sisu_www/cas
-

(B) to set up the search form

(i) copy estseek.cgi to your cgi directory and set file permissions to 755:

sudo cp -vi /usr/lib/estraier/estseek.cgi /usr/lib/cgi-bin

sudo chmod -v 755 /usr/lib/cgi-bin/estseek.cgi

sudo cp -v /usr/share/hyperestraier/estseek.* /usr/lib/cgi-bin

[see estraier documentation for paths]

(ii) edit estseek.conf, with attention to the lines starting ‘indexname:’ and ‘replace:’:

indexname: /home/ralph/sisu_www/casket

replace: $^{f}ile:///home/ralph/sisu\_www\{!\}$

replace: /index.html?${{!}}/

(C) to test using webrick, start webrick:

sisu -W

and try open the url: <http://localhost:8081/cgi-bin/estseek.cgi>

29. sisu_webrick

29.1 Name

SiSU - Structured information, Serialized Units - a document publishing system

29.2 Synopsis

sisu_webrick [port]

or

sisu -W [port]

29.3 Description

sisu_webrick is part of **SiSU** (man sisu) sisu_webrick starts **Ruby** 's Webrick web-server and points it to the directories to which **SiSU** output is written, providing a list of these directories (assuming **SiSU** is in use and they exist).

The default port for sisu_webrick is set to 8081, this may be modified in the yaml file: ~ /.sisu/sisurc.yml a sample of which is provided as /etc/sisu/sisurc.yml (or in the equivalent directory on your system).

29.4 Summary of man page

sisu_webrick, may be started on it's own with the command: sisu_webrick [port] or using the sisu command with the -W flag: sisu -W [port]

where no port is given and settings are unchanged the default port is 8081

29.5 Document processing command flags

sisu -W [port] starts **Ruby** Webrick web-server, serving **SiSU** output directories, on the port provided, or if no port is provided and the defaults have not been changed in ~ /.sisu/sisurc.yaml then on port 8081

29.6 Further information

For more information on **SiSU** see: <http://www.jus.uio.no/sisu>

or man sisu

29.7 Author

Ralph Amissah ralph@amissah.com or ralph.amissah@gmail.com

29.8 SEE ALSO

sisu(1)

sisu_vim(7)

sisu(8)

30. Remote Source Documents

SiSU processing instructions can be run against remote source documents by providing the url of the documents against which the processing instructions are to be carried out. The remote **SiSU** documents can either be sisu marked up files in plaintext .sst or .ssm or; zipped sisu files, sisupod.zip or filename.ssp

.sst / .ssm - sisu text files

SiSU can be run against source text files on a remote machine, provide the processing instruction and the url. The source file and any associated parts (such as images) will be downloaded and generated locally.

```
sisu -3 http://[provide url to valid .sst or .ssm file]
```

Any of the source documents in the sisu examples page can be used in this way, see <http://www.jus.uio.no/sisu/SiSU/2.htm and use the url for the desired document.

NOTE: to set up a remote machine to serve **SiSU** documents in this way, images should be in the directory relative to the document source ../_sisu/image

sisupod - zipped sisu files

A sisupod is the zipped content of a sisu marked up text or texts and any other associated parts to the document such as images.

SiSU can be run against a sisupod on a (local or) remote machine, provide the processing instruction and the url, the sisupod will be downloaded and the documents it contains generated locally.

```
sisu -3 http://[provide url to valid sisupod.zip or .ssp file]
```

Any of the source documents in the sisu examples page can be used in this way, see <http://www.jus.uio.no/sisu/SiSU/2.htm and use the url for the desired document.

REMOTE DOCUMENT OUTPUT 1139

31. Remote Output

Once properly configured **SiSU** output can be automatically posted once generated to a designated remote machine using either rsync, or scp.

In order to do this some ssh authentication agent and keychain or similar tool will need to be configured. Once that is done the placement on a remote host can be done seamlessly with the -r (for scp) or -R (for rsync) flag, which may be used in conjunction with other processing flags, e.g.

```
sisu -3R sisu_remote.sst
```

31.1 commands

-R [filename/wildcard]
copies sisu output files to remote host using rsync. This requires that sisurc.yml has been provided with information on hostname and username, and that you have your "keys" and ssh agent in place. Note the behavior of rsync different if -R is used with other flags from if used alone. Alone the rsync –delete parameter is sent, useful for cleaning the remote directory (when -R is used together with other flags, it is not). Also see -r

-r [filename/wildcard]
copies sisu output files to remote host using scp. This requires that sisurc.yml has been provided with information on hostname and username, and that you have your "keys" and ssh agent in place. Also see -R

31.2 configuration

[expand on the setting up of an ssh-agent / keychain]

32. Remote Servers

As **SiSU** is generally operated using the command line, and works within a Unix type environment, **SiSU** the program and all documents can just as easily be on a remote server, to which you are logged on using a terminal, and commands and operations would be pretty much the same as they would be on your local machine.

DOWNLOAD 1151

33. Download SiSU - Linux/Unix

SiSU Current Version - Linux/Unix

Source (tarball tar.gz)

Download the latest version of **SiSU** (and **SiSU** markup samples):[65]

- sisu_0.59.1.orig.tar.gz (of 2007-09-23:38/7) [66]

cgi generated sample search form

order results on files of the same title, in multiple files (with different filenames)

postgresql, character case sensitivity, control, on/off

tail decoration, gplv3 and sisu info

texinfo/info (pinfo) module starts to do something vaguely useful again [not a much used module, testing required]

print XML rendition of document structure to screen -T

help on environment visited, sisu -V

sisurc.yml default, color set to true [apologies if this causes anyone any inconvenience, it is configurable in sisurc.yml]

help and man pages, some work man(8) related

sisu-install (install ruby rant script renamed) and permissions set to executable

- sisu-markup-samples_1.0.8.orig.tar.gz (of 2007-08-19:33/7) [67]

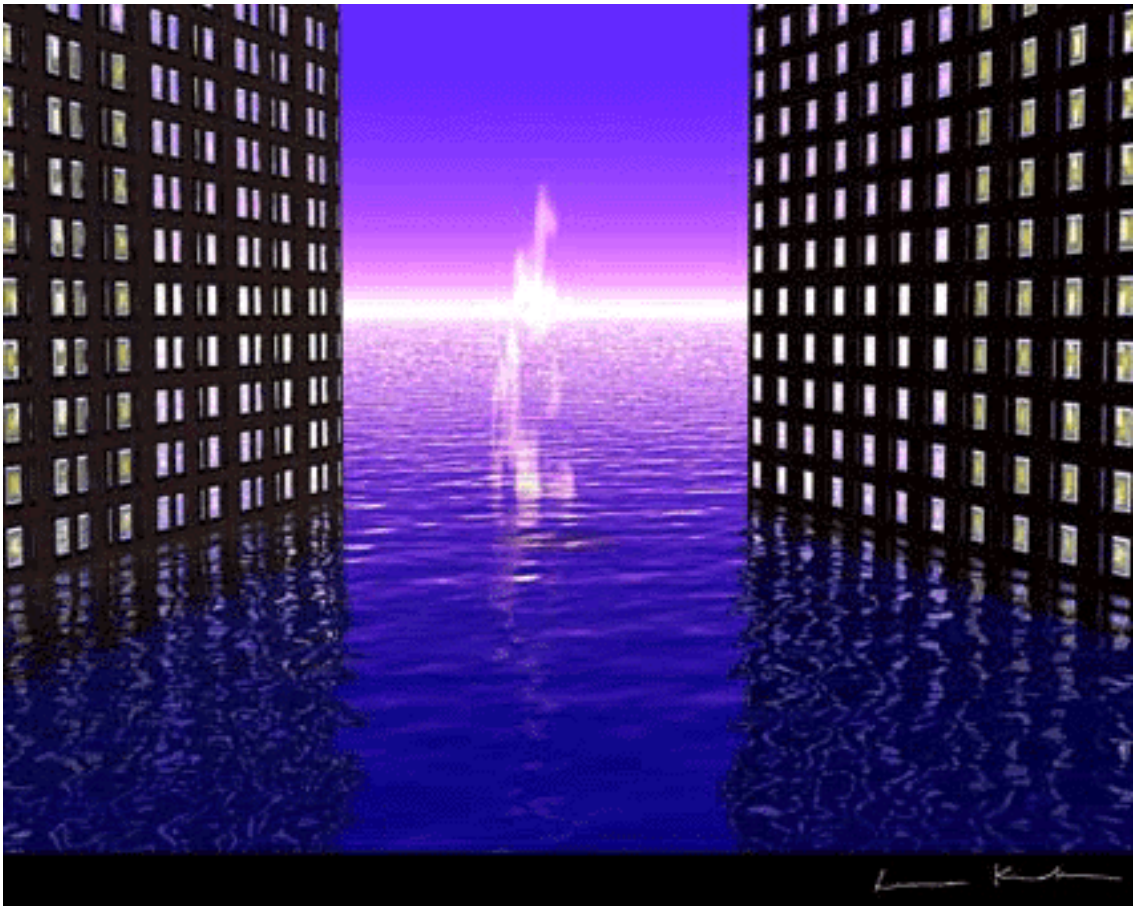
For installation notes see <http://www.jus.uio.no/sisu/SiSU/download.html>

For changelogs see <http://www.jus.uio.no/sisu/SiSU/changelog.html>

[65] Breakage and Fixes Report <http://www.jus.uio.no/sisu/SiSU/breakage_and_fixes.html>

[66] <http://www.jus.uio.no/sisu/archive/pool/main/s/sisu/sisu_0.59.1.orig.tar.gz>

[67] <http://www.jus.uio.no/sisu/archive/pool/non-free/s/sisu-markup-samples/sisu-markup-samples_1.0.7.orig.tar.gz>



Tulva, by Leena Krohn [68]

Git (source control management)

Git repository currently at:

- git clone git://sisudoc.org/git/sisu/
- <http://search.sisudoc.org/cgi-bin/gitweb.cgi?p=sisu;a=summary>

Debian

This section contains information on the latest **SiSU** release. For installation notes see <http://www.jus.uio.no/sisu/sisu_download/installation.html>

SiSU is updated fairly regularly in **Debian** testing and unstable, and should be available therefrom.

To add this archive, should you still choose to do so, add the following lines to your /etc/apt/sources.list

[68] Reproduced with the kind permission of author and artist Leena Krohn, <http://www.kaapeli.fi/krohn> Tulva is from the work Sphinx or Robot <http://www.jus.uio.no/sisu/sphinx_or_robot.leena_krohn.1996> other works available online include Tainaron <http://www.jus.uio.no/sisu/tainaron.leena_krohn.1998>, these two works can be found in the book sample section

deb http://www.jus.uio.no/sisu/archive unstable main non-free
deb-src http://www.jus.uio.no/sisu/archive unstable main non-free

Source

- sisu_0.59.1.orig.tar.gz [69]
- sisu_0.59.1-1.diff.gz [70]
- sisu_0.59.1-1.dsc [71]

Debs

- sisu_0.59.1-1_all.deb [72]
- sisu-complete_0.59.1-1_all.deb [73]
- sisu-pdf_0.59.1-1_all.deb [74]
- sisu-postgresql_0.59.1-1_all.deb [75]
- sisu-sqlite_0.59.1-1_all.deb [76]

For changelogs see:

<http://www.jus.uio.no/sisu/SiSU/changelog.html>

<http://www.jus.uio.no/sisu/sisu_changelog/changelog.html>

non-free

Book markup samples have been moved to non-free as the substantive text of the documents

http://www.jus.uio.no/sisu/SiSU/2.html#sample

[69] <http://www.jus.uio.no/sisu/archive/pool/main/s/sisu/sisu_0.59.1.orig.tar.gz>

[70] <http://www.jus.uio.no/sisu/archive/pool/main/s/sisu/sisu_0.59.1-1.diff.gz>

[71] <http://www.jus.uio.no/sisu/archive/pool/main/s/sisu/sisu_0.59.1-1.dsc>

[72] <http://www.jus.uio.no/sisu/archive/pool/main/s/sisu/sisu_0.59.1-1_all.deb>
sisu, the base code, (the main package on which the others depend), without any dependencies other than ruby (and for convenience the ruby webrick web server), this generates a number of types of output on its own, other packages provide additional functionality, and have their dependencies
Depends: ruby (¿=1.8.2), libwebrick-ruby
Recommends: sisu-pdf, sisu-sqlite, sisu-postgresql, sisu-examples, vim-sisu, librmagick-ruby, trang, tidy, libtidy, librexml-ruby, zip, unzip, openssl

[73] <http://www.jus.uio.no/sisu/archive/pool/main/s/sisu/sisu-complete_0.59.1-1_all.deb>
a package that pulls in other packages to build the whole of sisu (excluding sisu-examples)
Depends: ruby (¿=1.8.2), sisu, sisu-pdf, sisu-postgresql, sisu-remote, sisu-sqlite, vim-sisu
Recommends: sisu-examples

[74] <http://www.jus.uio.no/sisu/archive/pool/main/s/sisu/sisu-pdf_0.59.1-1_all.deb>
dependencies used by sisu to produce pdf from LaTeX generated
Depends: sisu, tetex-bin, tetex-extra, latex-ucs
Suggests: evince, xpdf

[75] <http://www.jus.uio.no/sisu/archive/pool/main/s/sisu/sisu-postgresql_0.59.1-1_all.deb>
dependencies used by sisu to populate postgresql database (further configuration is necessary)
Depends: sisu, postgresql-8.1, libdbi-ruby, libdbm-ruby, libdbd-pg-ruby
Suggests: pgaccess, libdbd-pgsql, postgresql-contrib-8.1

[76] <http://www.jus.uio.no/sisu/archive/pool/main/s/sisu/sisu-sqlite_0.59.1-1_all.deb>
dependencies used by sisu to populate sqlite database
Depends: sisu, sqlite, libdbi-ruby, libdbm-ruby, libdbd-sqlite-ruby
Suggests: libdbd-sqlite

are available under the author or original publisher's license, and usually do not comply with the **Debian** Free Software Guidelines.

- sisu-markup-samples_1.0.8-1_all.deb [77]
- sisu-markup-samples_1.0.8-1.dsc [78]

For changelogs see:

<http://www.jus.uio.no/sisu/SiSU/changelog_markup_samples.html>

<http://www.jus.uio.no/sisu/sisu_markup_samples_changelog/changelog_markup_samples.html>

RPM

The RPM is generated Alien[79] from two deb packages (sisu and sisu-doc). Dependencies are not handled, not even that of the essential **Ruby** .

- sisu-0.59.1-2.noarch.rpm [80]
- sisu-0.59.1-2.noarch.rpm [81]
- sisu-markup-samples_1.0.8.orig-2.noarch.rpm [82]

sudo rpm -i [package name]

[77] <http://www.jus.uio.no/sisu/archive/pool/non-free/s/sisu-markup-samples/sisu-markup-samples_1.0.8-1_all.deb>
marked up documents and other examples related to sisu, a larger package containing a number of texts
Depends: sisu

[78] <http://www.jus.uio.no/sisu/archive/pool/non-free/s/sisu-markup-samples/sisu-markup-samples_1.0.8-1.dsc>

[79] <http://www.kitenet.net/programs/alien/>

[80] <http://www.jus.uio.no/sisu/pkg/rpm/sisu-0.59.1-2.noarch.rpm>
untested, created using: alien -r sisu_0.59.1-1_all.deb

[81] <http://www.jus.uio.no/sisu/pkg/rpm/sisu-doc-0.59.1-2.noarch.rpm>
untested, created using: alien -r sisu-doc_0.59.1-1_all.deb

[82] <http://www.jus.uio.no/sisu/pkg/rpm/sisu-markup-samples_1.0.6.orig-2.noarch.rpm>
untested, created using alien

INSTALLATION 1206

34. Installation

See the download pages [83] for information related to installation.

34.1 Debian

SiSU is developed on **Debian** , and packages are available for **Debian** that take care of the dependencies encountered on installation.

The package is divided into the following components:

sisu , the base code, (the main package on which the others depend), without any dependencies other than ruby (and for convenience the ruby webrick web server), this generates a number of types of output on its own, other packages provide additional functionality, and have their dependencies

sisu-complete , a dummy package that installs the whole of greater sisu as described below, apart from sisu-examples

sisu-pdf , dependencies used by sisu to produce pdf from LaTeX generated

sisu-postgresql , dependencies used by sisu to populate postgresql database (further configuration is necessary)

sisu-remote , dependencies used to place sisu output on a remote server (further configuration is necessary)

sisu-sqlite , dependencies used by sisu to populate sqlite database

sisu-markup-samples , sisu markup samples and other miscellany (under **Debian** Free Software Guidelines non-free)

SiSU is available off Debian Unstable and Testing [84] install it using apt-get, aptitude or alternative **Debian** install tools. **SiSU** is currently comprised of eight packages.

Initial packaging is done here and to get the latest version of **SiSU** available you may add the following line(s) to your sources list:

```
#/etc/apt/sources.list
deb http://www.jus.uio.no/sisu/archive unstable main non-free
deb-src http://www.jus.uio.no/sisu/archive unstable main non-free
```

The non-free section is for sisu markup samples provided, which contain authored works the substantive text of which cannot be changed, and which as a result do not meet the debian free software guidelines.

On **Debian** there is little more to know beyond how to install software on **Debian** using apt, aptitude or synaptic.

```
#Using aptitude:
  aptitude update
  aptitude install sisu-complete sisu-markup-samples
```

[83] <http://www.jus.uio.no/sisu/SiSU/download>
<http://www.jus.uio.no/sisu/sisu_download>

[84] <http://packages.qa.debian.org/s/sisu.html>

Using apt-get
apt-get update
apt get install sisu-complete sisu-examples

34.2 Other Unix / Linux

A source tarball or an rpms built using alien are available, (however dependencies have not been tested). **SiSU** is first packaged and tested with dependency handling for **Debian** . [85] Information on dependencies configured for **Debian** is provided as this may be of assistance.

34.2.1 source tarball

installation with provided install script

To install **SiSU** , in the root directory of the unpacked **SiSU** as root type:[86]

ruby install

Once installed see man 8 sisu for information on additional programs that sisu makes use of.

Further notes on install script.

The install script is prepared using Rant, and a Rantfile is provided,[87] with more comprehensive install options, and post install and setup configuration and generation of first test file, if you have installed Stefan Lang's Rant [88] installed. While in the package directory, type: rant help, or rant -T, or to install **SiSU** as root, type:

install is an install script prepared using Stefan Lang's Rant [89] It should work whether you have previously installed Rant or not. It has fairly comprehensive install options, and can do some post install and setup configuration and generation of first test file. For options type:

ruby install -T

To install as root type:

ruby install

For a minimal install type:

ruby install base

installation with setup.rb

setup.rb [90] is provided the package and will install **SiSU** [91] installation is a 3 step process[92] the

[85] Notes on dependencies are provided in the section that follows
[86] This makes use of rant and the provided Rantfile. Note however, that additional external package dependencies, such as tetex-extra are not taken care of for you.
[87] a Rantfile has been configured to do post installation setup
[88] <http://make.rubyforge.org/> <http://make.rubyforge.org/> <http://rubyforge.org/frs/?group_id=615>
[89] <http://make.rubyforge.org/> <http://make.rubyforge.org/> <http://rubyforge.org/frs/?group_id=615>

following string assumes you are in the package directory and that you have root as sudo:

```
ruby setup.rb config  ruby setup.rb setup  sudo ruby setup.rb install
```

installation of rpm

The RPM is generated from the source file using Alien.[93] Dependencies are not handled, not even that of the essential **Ruby** .

[90] <http://i.loveruby.net/en/projects/setup/>

[91] Minero Aoki
<http://i.loveruby.net/en/projects/setup/doc/>

[92] Installation instructions
<http://i.loveruby.net/en/projects/setup/doc/usage.html>

[93] <http://www.kitenet.net/programs/alien/>

35. SiSU Components, Dependencies and Notes

The dependency lists are from the **Debian** control file for **SiSU** version 0.36, and may assist in building **SiSU** on other distributions.

35.1 sisu

- the base code, (the main package on which the others depend), without any dependencies other than ruby (and for convenience the ruby webrick web server), this generates a number of types of output on its own, other packages provide additional functionality, and have their dependencies

Depends: on ruby (¿=1.8.2), libwebrick-ruby

Recommends: sisu-pdf, sisu-sqlite, sisu-postgresql, sisu-examples, librmagick-ruby, trang, tidy, libtidy, librexml-ruby, zip, unzip, openssl

initialise directory

    sisu -CC

html

    sisu -hv [filename/wildcard]

    sisu -Hv [filename/wildcard]

LaTeX (but sisu-pdf dependencies required to convert that to pdf)

    sisu -pv [filename/wildcard]

plain text Unix with footnotes

    sisu -av [filename/wildcard]

plain text Dos with footnotes

    sisu -Av [filename/wildcard]

plain text Unix with endnotes

    sisu -ev [filename/wildcard]

plain text Dos with endnotes

    sisu -Ev [filename/wildcard]

openoffice odt

    sisu -ov [filename/wildcard]

xhtml

    sisu -bv [filename/wildcard]

XML SAX

    sisu -xv [filename/wildcard]

XML DOM

sisu -Xv [filename/wildcard]

wordmap (a rudimentary index of content)

sisu -wv [filename/wildcard]

document content certificate

sisu -Nv [filename/wildcard]

placement of sourcefile in output directory

sisu -sv [filename/wildcard]

creation of source tarball with images, and placement of source tarball in ouput directory

sisu -Sv [filename/wildcard]

manifest of output produced (polls output directory and provides links to existing output)

sisu -yv [filename/wildcard]

url for output files -u -U

sisu -uv[and other flags] [filename/wildcard]

sisu -Uv [filename/wildcard]

toggle screen colour

sisu -cv[and processing flags] [filename/wildcard]

verbose mode

sisu -v[and processing flags] [filename/wildcard]

sisu -V[and processing flags] [filename/wildcard]

quiet mode

sisu -q[and processing flags] [filename/wildcard]

maintenance mode, intermediate files kept -M

sisu -Mv[and other flags] [filename/wildcard]

[the -v is for verbose]

start the webrick server

sisu -W

35.2 sisu-complete

- a dummy package that installs the whole **SiSU** , apart from sisu-examples

Depends: ruby (¿=1.8.2), sisu, sisu-pdf, sisu-postgresql, sisu-remote, sisu-sqlite

Recommends: sisu-examples

35.3 sisu-examples

- installs sisu markup samples and other miscelleny

Depends: sisu

35.4 sisu-pdf

- dependencies used by sisu to produce pdf from LaTeX generated

Depends: sisu, tetex-bin, tetex-extra, latex-ucs

Suggests: evince, xpdf

converts sisu LaTeX produced to pdf

```
sisu -pv [filename/wildcard]
```

[the -v is for verbose]

35.5 sisu-postgresql

- dependencies used by sisu to populate postgresql database (further configuration is necessary)

Depends: sisu, postgresql-8.1, libdbi-ruby, libdbm-ruby, libdbd-pg-ruby

Suggests: pgaccess, libdbd-pgsql, postgresql-contrib-8.1

installs dependencies for sisu to work with and populate postgresql database

create database

```
sisu -Dv createall
```

drop database

```
sisu -Dv dropall
```

import content

```
sisu -Div [filename/wildcard]
sisu -Dv import [filename/wildcard]
```

update content

```
sisu -Duv [filename/wildcard]
sisu -Dv update [filename/wildcard]
```

[the -v is for verbose]

The following are available without installation of the sisu-postgresql component, but are of interest in this context

generate a sample database query form for use with webserver on port 80

```
sisu -F
```

or for use with webrick server

    sisu -F webrick

to start webrick server

    sisu -W

35.6 sisu-remote

- dependencies used to place sisu output on a remote server (further configuration is necessary)

scp

    sisu -vr[and processing flags] [filename/wildcard]

rsync

    sisu -vR[and processing flags] [filename/wildcard]

[the -v is for verbose]

Depends: sisu, rsync, openssh-client|lsh-client, keychain

35.7 sisu-sqlite

- dependencies used by sisu to populate sqlite database

Depends: sisu, sqlite, libdbi-ruby, libdbm-ruby, libdbd-sqlite-ruby

Suggests: libdbd-sqlite

installs dependencies for sisu to work with and populate sqlite database

create database

    sisu -dv createall

drop database

    sisu -dv dropall

update content

    sisu -div [filename/wildcard]

    sisu -dv import [filename/wildcard]

update content

    sisu -duv [filename/wildcard]

    sisu -dv update [filename/wildcard]

[the -v is for verbose]

The following are available without installation of the sisu-sqlite component, but are of interest in this context

generate a sample database query form for use with webserver on port 80

```
    sisu -F
or for use with webrick server
    sisu -F webrick
to start webrick server
    sisu -W
```

36. Quickstart - Getting Started Howto

36.1 Installation

Installation is currently most straightforward and tested on the **Debian** platform, as there are packages for the installation of sisu and all requirements for what it does.

36.1.1 Debian Installation

SiSU is available directly from the **Debian** Sid and testing archives (and possibly Ubuntu), assuming your /etc/apt/sources.list is set accordingly:

```
aptitude update
aptitude install sisu-complete
```

The following /etc/apt/sources.list setting permits the download of additional markup samples:

```
#/etc/apt/sources.list
deb http://ftp.fi.debian.org/debian/ unstable main non-free contrib
deb-src http://ftp.fi.debian.org/debian/ unstable main non-free contrib
d
```

The aptitude commands become:

```
aptitude update
aptitude install sisu-complete sisu-markup-samples
```

If there are newer versions of **SiSU** upstream of the **Debian** archives, they will be available by adding the following to your /etc/apt/sources.list

```
#/etc/apt/sources.list
deb http://www.jus.uio.no/sisu/archive unstable main non-free
deb-src http://www.jus.uio.no/sisu/archive unstable main non-free
```

repeat the aptitude commands

```
aptitude update
aptitude install sisu-complete sisu-markup-samples
```

Note however that it is not necessary to install sisu-complete if not all components of sisu are to be used. Installing just the package sisu will provide basic functionality.

36.1.2 RPM Installation

RPMs are provided though untested, they are prepared by running alien against the source package, and against the debs.

They may be downloaded from:

http://www.jus.uio.no/sisu/SiSU/download.html#rpm

as root type:

rpm -i [rpm package name]

36.1.3 Installation from source

To install **SiSU** from source check information at:

http://www.jus.uio.no/sisu/SiSU/download.html#current

- download the source package
- Unpack the source

Two alternative modes of installation from source are provided, setup.rb (by Minero Aoki) and a rant(by Stefan Lang) built install file, in either case: the first steps are the same, download and unpack the source file:

For basic use **SiSU** is only dependent on the programming language in which it is written **Ruby** , and **SiSU** will be able to generate html, various XMLs, including ODF (and will also produce LaTeX). Dependencies required for further actions, though it relies on the installation of additional dependencies which the source tarball does not take care of, for things like using a database (postgresql or sqlite)[94] or converting LaTeX to pdf.

setup.rb

This is a standard ruby installer, using setup.rb is a three step process. In the root directory of the unpacked **SiSU** as root type:

```
ruby setup.rb config
ruby setup.rb setup
#[and as root:]
ruby setup.rb install
```

further information on setup.rb is available from:

＜http://i.loveruby.net/en/projects/setup/＞

＜http://i.loveruby.net/en/projects/setup/doc/usage.html＞

“install”

The “install” file provided is an installer prepared using “rant”. In the root directory of the unpacked **SiSU** as root type:

ruby install base

or for a more complete installation:

ruby install

or

ruby install base

This makes use of Rant (by Stefan Lang) and the provided Rantfile. It has been configured to do post installation setup setup configuration and generation of first test file. Note however, that additional external package dependencies, such as tetex-extra are not taken care of for you.

Further information on “rant” is available from:

<http://make.rubyforge.org/>

<http://rubyforge.org/frs/?group_id=615>

For a list of alternative actions you may type:

ruby install help

ruby install -T

36.2 Testing SiSU, generating output

To check which version of sisu is installed:

sisu -v

Depending on your mode of installation one or a number of markup sample files may be found either in the directory:

...

or

...

change directory to the appropriate one:

cd /usr/share/doc/sisu/sisu_markup_samples/dfsg

36.2.1 basic text, plaintext, html, XML, ODF

Having moved to the directory that contains the markup samples (see instructions above if necessary), choose a file and run sisu against it

sisu -NhwoabxXyv free_as_in_freedom.rms_and_free_software.sam_williams.sst

this will generate html including a concordance file, opendocument text format, plaintext, XHTML and various forms of XML, and OpenDocument text

36.2.2 LaTeX / pdf

Assuming a LaTeX engine such as tetex or texlive is installed with the required modules (done automatically on selection of sisu-pdf in **Debian**)

Having moved to the directory that contains the markup samples (see instructions above if necessary), choose a file and run sisu against it

sisu -pv free_as_in_freedom.rms_and_free_software.sam_williams.sst

sisu -3 free_as_in_freedom.rms_and_free_software.sam_williams.sst

should generate most available output formats: html including a concordance file, opendocument text format, plaintext, XHTML and various forms of XML, and OpenDocument text and pdf

[94] There is nothing to stop MySQL support being added in future.

36.2.3 relational database - postgresql, sqlite

Relational databases need some setting up - you must have permission to create the database and write to it when you run sisu.

Assuming you have the database installed and the requisite permissions

sisu –sqlite –recreate

sisu –sqlite -v –import free_as_in_freedom.rms_and_free_software.sam_williams.sst

sisu –pgsql –recreate

sisu –pgsql -v –import free_as_in_freedom.rms_and_free_software.sam_williams.sst

36.3 Getting Help

36.3.1 The man pages

Type:

man sisu

The man pages are also available online, though not always kept as up to date as within the package itself:

- sisu.1 [95]
- sisu.8 [96]
- man directory [97]

36.3.2 Built in help

sisu –help

sisu –help –env

sisu –help –commands

sisu –help –markup

36.3.3 The home page

<http://www.jus.uio.no/sisu>

<http://www.jus.uio.no/sisu/SiSU>

36.4 Markup Samples

A number of markup samples (along with output) are available off:

[95] <http://www.jus.uio.no/sisu/man/sisu.1>
[96] <http://www.jus.uio.no/sisu/man/sisu.8>
[97] <http://www.jus.uio.no/sisu/man>

<http://www.jus.uio.no/sisu/SiSU/2.html>

Additional markup samples are packaged separately in the file:

*

On **Debian** they are available in non-free[98] to include them it is necessary to include non-free in your /etc/apt/source.list or obtain them from the sisu home site.

[98] the **Debian** Free Software guidelines require that everything distributed within **Debian** can be changed - and the documents are authors' works that while freely distributable are not freely changeable.

HOWTO

1464

37. Getting Help

An online manual of sorts should be available at:

<http://www.jus.uio.no/sisu_manual/>

The manual pages provided with **SiSU** are also available online, and there is an interactive help, which is being superseded by the man page, and possibly some document which contains this component.

37.1 SiSU "man" pages

If **SiSU** is installed on your system usual man commands should be available, try:

man sisu

The **SiSU** man pages can be viewed online at:[99]

An online version of the sisu man page is available here:

- various sisu man pages [100]
- sisu.1 [101]
- sisu.8 [102]
- sisu_examples.1 [103]
- sisu_webrick.1 [104]

37.2 SiSU built-in help

sisu –help

sisu –help [subject]

sisu –help env [for feedback on the way your system is setup with regard to sisu]

sisu -V [same as above command]

sisu –help commands

sisu –help markup

37.3 Command Line with Flags - Batch Processing

Running sisu (alone without any flags, filenames or wildcards) brings up the interactive help, as does any sisu command that is not recognised.

In the data directory run sisu -mh filename or wildcard eg. "sisu -h cisg.sst" or "sisu -h *.{sst,ssm}" to produce html version of all documents.

[99] generated from source using rman
<http://polyglotman.sourceforge.net/rman.html>
With regard to **SiSU** man pages the formatting generated for markup syntax is not quite right, for that you might prefer the links under:
<http://www.jus.uio.no/sample>

[100] <http://www.jus.uio.no/sisu/man/>

[101] <http://www.jus.uio.no/sisu/man/sisu.1.html>

[102] <http://www.jus.uio.no/sisu/man/sisu.8.html>

[103] <http://www.jus.uio.no/sisu/man/sisu_examples.1.html>

[104] <http://www.jus.uio.no/sisu/man/sisu_webrick.1.html>

38. Setup, initialisation

38.1 initialise output directory

Images, css files for a document directory are copied to their respective locations in the output directory.

while within your document markup/preparation directory, issue the following command

sisu -CC

38.1.1 Use of search functionality, an example using sqlite

SiSU can populate PostgreSQL and Sqlite databases and provides a sample search form for querying these databases.

This note provides an example to get you started and will use sqlite

It is necessary to:

(1) make sure the required dependencies have been installed

(2) have a directory with sisu markup samples that is writable

(3) use sisu to create a database

(4) use sisu tp populate a database

(5) use sisu to start the webrick (httpd) server

(6) use sisu to create a search form

(7) copy the search form to the cgi directory

(8) open up the form in your browser

(9) query the database using the search form

(1) make sure the required dependencies have been installed

if you use **Debian** , the following command will install the required dependencies

aptitude install sisu-sqlite

(2) have a directory with sisu markup samples that is writable

ideally copy the sisu-examples directory to your home directory (because the directory in which you run this example should be writable)

cp -rv /usr/share/sisu-examples/sample/document_samples_sisu_markup ~ /.

(3) use sisu to create an sqlite database

within the sisu-examples directory

sisu -dv createall

(4) use sisu tp populate a database with some text

within the sisu-examples directory

sisu -div free_*.sst

or

sisu -dv import free_*.sst debian_constitution_v1.2.sst debian_social_contract_v1.1.sst gpl2.fsf.sst

(5) use sisu to start the webrick (httpd) server (if it has not already been started):

sisu -W

(6) use sisu to create a search form (for use with the webrick server, and your sample documents)

within the sisu-examples directory

sisu -F webrick

#here i run into a problem, you are working from a read only #directory..., not my usual mode of operation, to complete the example #the following is necessary sudo touch sisu_sqlite.cgi sisu_pgsql.cgi sudo -P chown $USER sisu_sqlite.cgi sisu_pgsql.cgi

#now this should be possible: sisu -F webrick

(7) copy the search form to the cgi directory

the string should be provided as output from the previous command

sudo cp -vi /usr/share/sisu-examples/sample/document_samples_sisu_markup/sisu_sqlite.cgi /usr/lib/cgi-bin

sudo chmod -v 755 /usr/lib/cgi-bin/sisu_sqlite.cgi

(8) open up the form in your browser and query it

url:

<http://localhost:8081/cgi-bin/sisu_sqlite.cgi>

or as instructed by command sisu -F webrick

(9) query the database using the search form

if there are other options in the dropdown menu select

document_samples_sisu_markup

and search for some text, e.g.:

aim OR project

- selecting the **index** radio button gives an index of results using the object numbers
- selecting the **text** radio button gives the content of the matched paragraphs with the match highlighted

(10) to start again with a new database

to start from scratch you can drop the database with the command

sisu -dv dropall

and go to step 3

to get to step 3 in one step with a single command
sisu -dv recreate
continue subsequent steps

38.2 misc

38.2.1 url for output files -u -U

sisu -uv[and other flags] [filename/wildcard]
sisu -Uv [filename/wildcard]

38.2.2 toggle screen color

sisu -cv[and processing flags] [filename/wildcard]

38.2.3 verbose mode

sisu -v[and processing flags] [filename/wildcard]
sisu -V[and processing flags] [filename/wildcard]

38.2.4 quiet mode

sisu -q[and processing flags] [filename/wildcard]

38.2.5 maintenance mode intermediate files kept -M

sisu -Mv[and other flags] [filename/wildcard]

38.2.6 start the webrick server

sisu -W

38.3 remote placement of output

configuration is necessary
scp
sisu -vr[and processing flags] [filename/wildcard]
rsync
sisu -vR[and processing flags] [filename/wildcard]

39. Configuration Files

Sample provided, on untarring the source tarball:

conf/sisu/sisurc.yaml

and on installation under:

/etc/sisu/sisurc.yaml

The following paths are searched:

./_sisu/sisurc.yaml

~ /.sisu/sisurc.yaml

./etc/sisu/sisurc.yaml

40. Markup

See sample markup provided on

<http://www.jus.uio.no/sisu>

<http://www.jus.uio.no/sisu_markup>

<http://www.jus.uio.no/sisu/SiSU>

in particular for each of the document output samples provided, the source document is provided as well

http://www.jus.uio.no/sisu/SiSU/2#books

on untarring the source tarball:

data/sisu-examples/sample/document_samples_sisu_markup/

or the same once source is installed (or sisu-examples) under:

/usr/share/sisu-examples/sample/document_samples_sisu_markup/

Some notes are contained within the man page, **man sisu** and within sisu help via the commands **sisu help markup** and **sisu help headers**

SiSU is for literary and legal text, also for some social science material. In particular it does not do formula, and is not particularly suited to technical documentation. Despite the latter caveat, some notes will be provided here and added to over time:

40.1 Headers

Headers @headername: provide information related to the document, this may relate to

1. how it is to be processed, such as whether headings are to be numbered, what skin is to be used and markup instructions, such as the document structure, or words to be made bold within the document

2. semantic information about the document including the dublin core

40.2 Font Face

Defaults are set. You may change the face to: bold, italics, underscore, strikethrough, ...

40.2.1 Bold

bold: [list of words that should be made bold within document]

bold line

!_ bold line

bold word or sentence

!{ bold word or sentence }!

{ bold word or sentence }

boldword or **boldword**

boldword or !boldword!

40.2.2 Italics

italics: [list of words that should be italicised within document]

italicise word or sentence

/{ italicise word or sentence }/

italicisedword

/italicisedword/

40.2.3 Underscore

underscore word or sentence

{ underscore word or sentence }

underscoreword

40.2.4 Strikethrough

~~strikethrough word or sentence~~

-{ strikethrough word or sentence }-

~~strikeword~~

-strikeword-

40.3 Endnotes

There are two forms of markup for endnotes, they cannot be mixed within the same document

here[105]

1. preferred endnote markup

here~{ this is an endnote }~

2. alternative markup equivalent, kept because it is possible to search and replace to get markup in existing texts such as Project Gutenberg

here~

~*thisisanendote*

[105] this is an endnote

40.4 Links

SiSU

{ **SiSU** }http://sisudoc.org

{sisu.png }http://sisudoc.org

{ tux.png 64x80 }image

SiSU [106]

{ **SiSU** }http://sisudoc.org

is equivalent to:

{ **SiSU** }http://sisudoc.org ~ { <http://sisudoc.org> }~

the same can be done with an image:

SiSU [108]

{ sisu.png "**SiSU** " }http://sisudoc.org

40.5 Number Titles

Set with the header @markup:

40.6 Line operations

Line Operations (marker placed at start of line)

!_ bold line

bold line

_1 indent paragraph one level

indent paragraph one level

_2 indent paragraph two steps

[106] <http://sisudoc.org>

[108] <http://sisudoc.org>

indent paragraph two steps

_* bullet paragraph

• bullet paragraph

{# number paragraph (see headers for numbering document headings)

1. number paragraph (see headers for numbering document headings)

_# number paragraph level 2 (see headers for numbering document headings)

a. number paragraph level 2 (see headers for numbering document headings)

40.7 Tables

Table markup sample

table{~ h c3; 26; 32; 32;

This is a table, column1
this would become row one of column two
column three of row one is here

column one row 2
column two of row two
column three of row two, and so on

column one row three
and so on
here

}table

Alternative form of table markup

{t~ h}

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
0	*	*	*	*	*	*	*
1	*	*	*	*			
2	-	*	*	*	*	*	
3	-	*	*	*	*	*	*
4	-			*	*	*	
5	*	*	*	*	*	*	*

40.8 Grouped Text

```
  5.times { puts 'Ruby' }
```

code{

```
  5.times { puts 'Ruby' }
```

}code

A Limerick

There was a young lady from Clyde,
who ate a green apple and died,
but the apple fermented inside the lamented,
and made cider inside her inside.

poem{

There was a young lady from Clyde,
who ate a green apple and died,
but the apple fermented inside the lamented,
and made cider inside her inside.

}

40.9 Composite Document

To import another document, the master document or importing document should be named filename.r3 (r for require)

<< { filename.sst }

<< { filename.ssi }

41. Change Appearance

41.1 Skins

"Skins" may be used to change various aspects related to the output documents appearance, including such things as the url for the home page on which the material will be published, information on the credit band, and for html documents colours and icons used in navigation bars. Skins are ruby files which permit changing of the default values set within the program for **SiSU** output.

There are a few examples provided, on untarring the source tarball:

conf/sisu/skin/doc/

data/sisu-examples/sample/document_samples_sisu_markup/_sisu/skin/doc

and on installation under:

/etc/sisu/skin/doc/

/usr/share/sisu-examples/sample/document_samples_sisu_markup/_sisu/skin/doc

The following paths are searched:

./_sisu/skin

~ /.sisu/skin

/etc/sisu/skin

Skins under the searched paths in a per document directory, a per directory directory, or a site directory, named:

doc [may be specified individually in each document]

dir [used if identifier part of name matches markup directory name]

site

It is usual to place all skins in the document directory, with symbolic links as required from dir or site directories.

41.2 CSS

The appearance of html and XML related output can be changed for an ouput collection directory by prepareing and placing a new css file in one of the sisu css directories searched in the sisu configuration path. These are located at:

../_sisu/css

~ /.sisu/css

and

/etc/sisu/css

The contents of the first directory found in the search path are copied to the corresponding sisu

output directory with the commnd:

sisu -CC

The **SiSU** standard css files for **SiSU** output are:

dom.css html.css html_tables.css index.css sax.css xhtml.css

A document may specify its own/bespoke css file using the css header.

css:

[expand]

EXTRACTS FROM THE README

1715

42. README

SiSU 0.55 2007w27/6 2007-07-07

Homepage: <http://www.jus.uio.no/sisu>

Description

SiSU is lightweight markup based document creation and publishing framework that is controlled from the command line. Prepare documents for **SiSU** using your text editor of choice, then use **SiSU** to generate various output document formats.

With minimal preparation of a plain-text (UTF-8) file using its native markup-syntax, **SiSU** produces: plain-text, HTML, XHTML, XML, ODF:ODT (Opendocument), LaTeX, PDF, and populates an SQL database (PostgreSQL or SQLite) in paragraph sized chunks so that document searches are done at this "atomic" level of granularity.

Outputs share a common citation numbering system, and any semantic meta-data provided about the document.

SiSU also provides concordance files, document content certificates and manifests of generated output.

SiSU takes advantage of well established open standard ways of representing text, and provides a bridge to take advantage of the strengths of each, while remaining simple. **SiSU** implements across document formats a "useful common feature set" [coming from a humanities, law, and possibly social sciences perspective, rather than technical or scientific writing] ... focus is primarily on content and data integrity rather than appearance, (though outputs in the various formats are respectable).

A vim syntax highlighting file and an ftplugin with folds for sisu markup is provided. Vim 7 includes syntax highlighting for **SiSU** .

man pages, and interactive help are provided.

Dependencies for various features are taken care of in sisu related packages. The package sisu-complete installs the whole of **SiSU** .

Additional document markup samples are provided in the package sisu-markup-samples which is found in the non-free archive the licenses for the substantive content of the marked up documents provided is that provided by the author or original publisher.

Homepage: <http://www.jus.uio.no/sisu>

SiSU - simple information structuring universe, is a publishing tool, document generation and management, (and search enabling) tool primarily for literary, academic and legal published works.

SiSU can be used for Internet, Intranet, local filesystem or cd publishing.

SiSU can be used directly off the filesystem, or from a database.

SiSU 's scalability, is be dependent on your hardware, and filesystem (in my case Reiserfs), and/or database Postgresql.

Amongst it's characteristics are:

- simple mnemonoic markup style,
- the ability to produce multiple output formats, including html, structured XML, LaTeX, pdf (via LaTeX), stream to a relational database whilst retaining document structure - Postgresql and Sqlite,
- that all share a common citation system (a simple idea from which much good), possibly most exciting, the following: if fed into a relational database (as it can be automatically), the document set is searchable, with results displayed at a paragraph level, or the possibility of an indexed display of documents in which the match is found together with a hyperlinked listing for each of each paragraph in which the match is found. In any event citations using this system (with or without the relational database) are relevant for all output formats.
- it is command line driven, and can be set up on a remote server
- Documents are marked up in **SiSU** syntax in your favourite editor. **SiSU** syntax may be regarded as a type of smart ascii - which in its basic form is simpler than the most elementary html. There is currently a syntax highlighter, and folding for Vim. Syntax highlighters for other editors are welcome.

Input files should be UTF-8

Once set up it is simple to use.

42.1 Online Information, places to look

<http://www.jus.uio.no/sisu>

Download Sources:

http://www.jus.uio.no/sisu/SiSU/download.html#current

http://www.jus.uio.no/sisu/SiSU/download.html#debian

42.2 Installation

NB. Platform is Unix / Linux.

42.2.1 Debian

If you use **Debian** use the **Debian** packages, check the information at:

http://www.jus.uio.no/sisu/SiSU/download.html#debian

(A) **SiSU** is available directly off the **Debian** archives for Sid and testing. It should necessary only to run as root:

aptitude update

aptitude install sisu-complete

(B) If there are newer versions of **SiSU** upstream of the **Debian** archives, they will be available by adding the following to your /etc/apt/sources.list

deb <http://www.jus.uio.no/sisu/archive> unstable main non-free

deb-src <http://www.jus.uio.no/sisu/archive> unstable main non-free

[the non-free line is for document markup samples, for which the substantive text is provided under the author or original publisher's license and which in most cases will not be debian free software guideline compliant]

Then as root run:

aptitude update

aptitude install sisu-complete

42.2.2 RPM

RPMs are provided though untested, they are prepared by running alien against the source package, and against the debs.

They may be downloaded from:

http://www.jus.uio.no/sisu/SiSU/download.html#rpm

42.2.3 Source package .tgz

Otherwise to install **SiSU** from source, check information at:

http://www.jus.uio.no/sisu/SiSU/download.html#current

alternative modes of installation from source are provided, setup.rb (by Minero Aoki), rake (by Jim Weirich) built install file, rant (by Stefan Lang) built install file,

Ruby is the essential dependency for the basic operation of **SiSU**

1. Download the latest source (information available) from:

http://www.jus.uio.no/sisu/SiSU/download.html#current

2. Unpack the source

Note however, that additional external package dependencies, such as texlive or postgresql should you desire to use it are not taken care of for you.

42.2.4 to use setup.rb

this is a three step process, in the root directory of the unpacked **SiSU** as root type:

ruby setup.rb config

ruby setup.rb setup

as root:

ruby setup.rb install

further information:

<http://i.loveruby.net/en/projects/setup/>

<http://i.loveruby.net/en/projects/setup/doc/usage.html>

42.2.5 to use install (prapared with "Rake")

Rake must be installed on your system:

<http://rake.rubyforge.org/>

<http://rubyforge.org/frs/?group_id=50>

in the root directory of the unpacked **SiSU** as root type:

```
rake
```

or

```
rake base
```

This makes use of Rake (by Jim Weirich) and the provided Rakefile

For a list of alternative actions you may type:

```
rake help
rake -T
```

42.2.6 to use install (prapared with "Rant")

(you may use the instructions above for rake substituting rant if rant is installed on your system, or you may use an independent installer created using rant as follows:)

in the root directory of the unpacked **SiSU** as root type:

```
ruby ./sisu-install
```

or

```
ruby ./sisu-install base
```

This makes use of Rant (by Stefan Lang) and the provided Rantfile. It has been configured to do post installation setup setup configuration and generation of first test file. Note however, that additional external package dependencies, such as tetex-extra are not taken care of for you.

further information:

<http://make.rubyforge.org/>

<http://rubyforge.org/frs/?group_id=615>

For a list of alternative actions you may type:

```
ruby ./sisu-install help
ruby ./sisu-install -T
```

42.3 Dependencies

Once installed see 'man 8 sisu' for some information on additional programs that sisu makes use of, and that you may need or wish to install. (this will depend on such factors as whether you want to generate pdf, whether you will be using **SiSU** with or without a database, ...) 'man sisu_markup-samples' may also be of interest if the sisu-markup-samples package has also been installed.

The information in man 8 may not be most up to date, and it is possible that more useful information can be gleaned from the following notes taken from the **Debian** control file (end edited), gives an idea of additional packages that **SiSU** can make use of if available, (the use/requirement of some of which are interdependent for specific actions by **SiSU**).

The following is from the debian/control file of sisu-0.58.2, which amongst other things provides the dependencies of sisu within **Debian** .

```
Package: sisu
Architecture: all
Depends: ruby (>= 1.8.2), libwebrick-ruby, unzip, zip
Conflicts: vim-sisu, sisu-vim, sisu-remote
Replaces: vim-sisu, sisu-vim
Recommends: sisu-pdf, sisu-sqlite, sisu-postgresql, librmagick-ruby, trang,
tidy, librexml-ruby, openssl, rsync, openssh-client | lsh-client, keychain,
hyperestraier, kdissert, vim-addon-manager
Suggests: rcs | cvs, lv, texinfo, pinfo
Package: sisu-complete
Depends: ruby (>= 1.8.4), sisu, sisu-pdf, sisu-postgresql, sisu-sqlite
Recommends: hyperestraier
Package: sisu-pdf
Architecture: all
Depends: sisu, texlive-latex-base, texlive-fonts-recommended,
texlive-latex-recommended, texlive-latex-extra
Suggests: evince, xpdf
Package: sisu-postgresql
Depends: sisu, postgresql-8.1, libdbi-ruby, libdbm-ruby, libdbd-pg-ruby
Suggests: pgaccess, libdbd-pgsql, postgresql-contrib-8.1
Package: sisu-sqlite
Depends: sisu, sqlite, libdbi-ruby, libdbm-ruby, libdbd-sqlite-ruby
Suggests: libdbd-sqlite
Package: sisu-markup-samples
Depends: sisu
Source: sisu
Section: text
Priority: optional
Maintainer: Ralph Amissah <ralph@amissah.com>
Build-Depends: debhelper (>= 5)
Standards-Version: 3.7.2
Package: sisu
Architecture: all
Depends: ruby (>= 1.8.2), ruby (<< 1.9), libwebrick-ruby, unzip, zip
Conflicts: vim-sisu, sisu-vim, sisu-remote
Replaces: vim-sisu, sisu-vim
Recommends: sisu-doc, sisu-pdf, sisu-sqlite, sisu-postgresql, hyperestraier, keychain, librmagick-ruby, librexml-ruby, openssl, openssh-client
| lsh-client, rsync, tidy, vim-addon-manager
```

```
Suggests: kdissert, lv, rcs | cvs, pinfo, texinfo, trang
Description: documents - structuring, publishing in multiple formats and search
 SiSU is a lightweight markup based, command line oriented, document
 structuring, publishing and search framework for document collections.
 .
 With minimal preparation of a plain-text, (UTF-8) file, using its native
 markup syntax in your text editor of choice, SiSU can generate various
 document formats (most of which share a common object numbering system for
 locating content), including plain text, HTML, XHTML, XML, OpenDocument text
 (ODF:ODT), LaTeX, PDF files, and populate an SQL database with objects
 (roughly paragraph-sized chunks) so searches may be performed and matches
 returned with that degree of granularity: your search criteria is met by these
 documents and at these locations within each document. Object numbering is
 particularly suitable for "published" works (finalized texts as opposed to
 works that are frequently changed or updated) for which it provides a fixed
 means of reference of content. Document outputs also share semantic meta-data
 provided.
 .
 SiSU also provides concordance files, document content certificates and
 manifests of generated output.
 .
 A vim syntax highlighting file and an ftplugin with folds for sisu markup is
 provided, as are syntax highlighting files for kate, kwrite, gedit and
 diakonos. Vim 7 includes syntax highlighting for SiSU.
 .
 man pages, and interactive help are provided.
 .
 Dependencies for various features are taken care of in sisu related packages.
 The package sisu-complete installs the whole of SiSU.
 .
 Additional document markup samples are provided in the package
 sisu-markup-samples which is found in the non-free archive the licenses for
 the substantive content of the marked up documents provided is that provided
 by the author or original publisher.
 .
 Homepage: <http://www.jus.uio.no/sisu>                                    1815
Package: sisu-complete
Architecture: all
Depends: ruby (>= 1.8.2), ruby (<< 1.9), sisu, sisu-doc, sisu-pdf, sisu-postgresql, sisu-sqlite
Recommends: hyperestraier
Description: installs all SiSU related packages
 This package installs SiSU and related packages that enable sisu to produce
 pdf and to populate postgresql and sqlite databases.
 .
 SiSU is a lightweight markup based document structuring, publishing and search
 framework for document collections.
 .
 See sisu for a description of the package.
 .
 Homepage: <http://www.jus.uio.no/sisu>                                    1816
Package: sisu-doc
Architecture: all
Depends: sisu
```

Recommends: sisu-pdf, sisu-postgresql, sisu-sqlite
Description: sisu manual and other documentation for sisu
 Multiple file formats generated output of sisu documentation generated from
 sisu markup source documents included in the main package
 .
 SiSU is a lightweight markup based document structuring, publishing and search
 framework for document collections.
 .
 Homepage: <http://www.jus.uio.no/sisu> 1817
Package: sisu-pdf
Architecture: all
Depends: sisu, texlive-latex-base, texlive-fonts-recommended, texlive-latex-recommended, texlive-latex-extra
Recommends: sisu-doc
Description: dependencies to convert SiSU LaTeX output to pdf
 This package enables the conversion of SiSU LaTeX output to pdf.
 .
 SiSU is a lightweight markup based document structuring, publishing and search
 framework for document collections.
 .
 Homepage: <http://www.jus.uio.no/sisu> 1818
Package: sisu-postgresql
Architecture: all
Depends: sisu, libdbd-pg-ruby, libdbi-ruby, libdbm-ruby, postgresql
Recommends: sisu-doc, libfcgi-ruby
Suggests: postgresql-contrib
Description: SiSU dependencies for use with postgresql database
 This package enables SiSU to populate a postgresql database. This is done at
 an object/paragraph level, making granular searches of documents possible.
 .
 This relational database feature of SiSU is not required but provides
 interesting possibilities, including that of granular searches of documents
 for matching units of text, primarily paragraphs that can be displayed or
 identified by object citation number, from which an index of documents
 matched and each matched paragraph within them can be displayed.
 .
 SiSU is a lightweight markup based document structuring, publishing and search
 framework for document collections.
 .
 Homepage: <http://www.jus.uio.no/sisu> 1819
Package: sisu-sqlite
Architecture: all
Depends: sisu, sqlite, libdbd-sqlite-ruby, libdbi-ruby, libdbm-ruby
Recommends: sisu-doc, libfcgi-ruby
Description: SiSU dependencies for use with sqlite database
 This package enables SiSU to populate an sqlite database. This is done at an
 object/paragraph level, making granular searches of documents possible.
 .
 This relational database feature of SiSU is not required but provides
 interesting possibilities, including that of granular searches of documents
 for matching units of text, primarily paragraphs that can be displayed or
 identified by object citation number, from which an index of documents
 matched and each matched paragraph within them can be displayed.
 .

SiSU is a lightweight markup based document structuring, publishing and search framework for document collections.

.

Homepage: <http://www.jus.uio.no/sisu>

42.4 Quick start

Most of the installation should be taken care of by the aptitude or rant install. (The rant install if run in full will also test run the generation of the first document).

After installation of sisu-complete, move to the document samples directory

cd /usr/share/doc/sisu/sisu_markup_samples/dfsg

and run

sisu -3 free_as_in_freedom.rms_and_free_software.sam_williams.sst

or the same:

sisu -NhwpoabxXyv free_as_in_freedom.rms_and_free_software.sam_williams.sst

look at output results, see the "sisu_manifest" page created for the document

or to generate an online document move to a writable directory, as the file will be downloaded there and e.g.

sisu -3 <http://www.jus.uio.no/sisu/free_culture.lawrence_lessig/free_culture.lawrence_lessig.sst>

the database stuff is extra perhaps, the latex stuff could be considered extra perhaps but neither needs to be installed for most of sisu output to work

examine source document, vim has syntax support

gvim free_as_in_freedom.rms_and_free_software.sam_williams.sst

additional markup samples in

<http://www.jus.uio.no/sisu/SiSU/2.html>

For help

man sisu

or

sisu –help

e.g.

for the way sisu "sees/maps" your system

sisu –help env

for list of commands and so on

sisu –help commands

42.5 Configuration files

The default configuration/setup is contained within the program and is altered by configuration settings in *etc*[sisu version]/sisurc.yml or in ~ /.sisu/sisurc.yml

- configuration file - a yaml file

  /etc/sisu/[sisu version]/sisurc.yml

  ~ /.sisu/sisurc.yml

- directory structure - setting up of output and working directory.

* skins - changing the appearance of a project, directory or individual documents within ~ /.sisu/skin

  ~ /.sisu/skin/doc contains individual skins, with symbolic links from

  ~ /.sisu/skin/dir if the contents of a directory are to take a particular document skin.

- additional software - eg. Tex and LaTeX (tetex, tetex-base, tetex-extra on **Debian**), Postgresql, [sqlite], trang, tidy, makeinfo, ... none of which are required for basic html or XML processing.

- if you use Vim as editor there is a syntax highlighter and fold resource config file for **SiSU** . I hope more syntax highlighters follow.

There are post installation steps (which are really part of the overall installation)

sisu -C in your marked up document directory, should do some auto-configuring provided you have the right permissions for the output directories. (and provided the output directories have already been specified if you are not using the defaults).

42.6 Use General Overview

Documents are marked up in **SiSU** syntax and kept in an ordinary text editable file, named with the suffix .sst, or .ssm

Marked up **SiSU** documents are usually kept in a sub-directory of your choosing

use the interactive help and man pages

  sisu –help

  man sisu

42.7 Help

interactive help described below, or man page:

  man sisu

  man 8 sisu

‘man sisu_markup-samples’ [if the sisu-markup-samples package is also installed]

Once installed an interactive help is available typing ‘sisu’ (without) any flags, and select an option:

sisu

alternatively, you could type e.g.

sisu –help commands

sisu –help env

sisu –help headers

sisu –help markup

sisu –help headings

etc.

for questions about mappings, output paths etc.

sisu –help env

sisu –help path

sisu –help directory

42.8 Directory Structure

Once installed, type:

sisu –help env

or

sisu -V

42.9 Configuration File

The defaults can be changed via **SiSU** ’s configure file sisurc.yml which the program expects to find in ./_sisu ~ /.sisu or /etc/sisu (searched in that order, stopping on the first one found)

42.10 Markup

See man pages.

man sisu

man 8 sisu

Once installed there is some information on **SiSU** Markup in its help:

sisu –help markup

and

sisu –help headers

Sample marked up document are provided with the download tarball in the directory:

./data/doc/sisu/sisu_markup_samples/dfsg

These are installed on the system usually at:

/usr/share/doc/sisu/sisu_markup_samples/dfsg

More markup samples are available in the package sisu-markup-samples

http://www.jus.uio.no/sisu/SiSU/download.html#sisu-markup-samples

Many more are available online off:

<http://www.jus.uio.no/sisu/SiSU/2.html>

42.11 Additional Things

There is syntax support for some editors provided (together with a README file) in

./data/sisu/conf/syntax

usually installed to:

/usr/share/sisu/conf/syntax

42.12 License

License: GPL 3 or later see the copyright file in

./data/doc/sisu

usually installed to:

/usr/share/doc/sisu

42.13 SiSU Standard

SiSU uses:

- Standard **SiSU** markup syntax,
- Standard **SiSU** meta-markup syntax, and the
- Standard **SiSU** object citation numbering and system

© Ralph Amissah 1997, current 2006 All Rights Reserved.

- however note the License section

CHANGELOG

./CHANGELOG

and see

<http://www.jus.uio.no/sisu/SiSU/changelog.html>

<http://www.jus.uio.no/sisu/SiSU/changelog_markup_samples.html>

EXTRACTS FROM MAN 8 SISU

1927

43. Post Installation Setup

43.1 Post Installation Setup - Quick start

After installation of sisu-complete, move to the document samples directory,

cd /usr/share/doc/sisu/sisu_markup_samples/dfsg

[this is not where you would normally work but provides sample documents for testing, you may prefer instead to copy the contents of that directory to a local directory before proceeding]

and in that directory, initialise the output directory with the command

sisu -CC

then run:

sisu -1 free_as_in_freedom.rms_and_free_software.sam_williams.sst

or the same:

sisu -NhwpoabxXyv free_as_in_freedom.rms_and_free_software.sam_williams.sst

look at output results, see the "sisu_manifest" page created for the document

for an overview of your current sisu setup, type:

sisu –help env

or

sisu -V

To generate a document from a remote url accessible location move to a writable directory, (create a work directory and cd into it) as the file will be downloaded there and e.g.

sisu -1 <http://www.jus.uio.no/sisu/gpl3.fsf/gpl3.fsf.sst>

sisu -3 <http://www.jus.uio.no/sisu/free_culture.lawrence_lessig/free_culture.lawrence_lessig.sst>

examine source document, vim has syntax highlighting support

gvim free_as_in_freedom.rms_and_free_software.sam_williams.sst

additional markup samples in

<http://www.jus.uio.no/sisu/SiSU/2.html>

it should also be possible to run sisu against sisupods (prepared zip files, created by running the command sisu -S [filename]), whether stored locally or remotely.

sisu -3 <http://www.jus.uio.no/sisu/free_culture.lawrence_lessig/sisupod.zip>

there is a security issue associated with the running of document skins that are not your own, so these are turned of by default, and the use of the following command, which switches on the associated skin is not recommended:

sisu -3 –trust <http://www.jus.uio.no/sisu/free_culture.lawrence_lessig/sisupod.zip>

For help

man sisu

sisu –help

sisu –help env for the way sisu "sees/maps" your system

sisu –help commands for list of commands and so on

43.2 Document markup directory

Perhaps the easiest way to begin is to create a directory for sisu marked up documents within your home directory, and copy the file structure (and document samples) provided in the document sample directory:

mkdir ~ /sisu_test

cd ~ /sisu_test

cp -a /usr/share/doc/sisu/sisu_markup_samples/dfsg/* ~ /sisu_test/.

Tip: the markup syntax examples may be of interest

<http://www.jus.uio.no/sisu/sample/>

Tip:

sisu -U [sisu markup filename]

should printout the different possible outputs and where sisu would place them.

Tip: if you want to toggle ansi color add

c

to your flags.

43.2.1 Configuration files

SiSU configuration file search path is:

./_sisu/sisurc.yaml

~ /.sisu/sisurc.yaml

/etc/sisu/sisurc.yaml

.%% **Debian** Installation Note

43.2.2 Debian INSTALLATION Note

It is best you see

http://www.jus.uio.no/sisu/SiSU/download.html#debian

for up the most up to date information.

notes taken from the **Debian** control file (end edited), gives an idea of additional packages that **SiSU** can make use of if available, (the use/requirement of some of which are interdependent for specific actions by **SiSU**):

Package: sisu

SiSU is a lightweight markup based, command line oriented, document structuring, publishing and search framework for document collections.

With minimal preparation of a plain-text, (UTF-8) file, using its native markup syntax in your text editor of choice, **SiSU** can generate various document formats (most of which share a common object numbering system for locating content), including plain text, HTML, XHTML, XML, OpenDocument text (ODF:ODT), LaTeX, PDF files, and populate an SQL database with objects (roughly paragraph-sized chunks) so searches may be performed and matches returned with that degree of granularity: your search criteria is met by these documents and at these locations within each document. Object numbering is particularly suitable for "published" works (finalized texts as opposed to works that are frequently changed or updated) for which it provides a fixed means of reference of content. Document outputs also share semantic meta-data provided.

SiSU also provides concordance files, document content certificates and manifests of generated output.

A vim syntax highlighting file and an ftplugin with folds for sisu markup is provided, as are syntax highlighting files for kate, kwrite, gedit and diakonos. Vim 7 includes syntax highlighting for **SiSU** .

man pages, and interactive help are provided.

Dependencies for various features are taken care of in sisu related packages. The package sisu-complete installs the whole of **SiSU** .

Additional document markup samples are provided in the package sisu-markup-samples which is found in the non-free archive the licenses for the substantive content of the marked up documents provided is that provided by the author or original publisher.

Homepage: <http://www.jus.uio.no/sisu>

43.2.3 Document Resource Configuration

sisu resource configuration information is obtained from sources (where they exist):

~ /.sisu/sisurc.yaml

/etc/sisu/[sisu version]/sisurc.yaml

sisu program defaults

43.2.4 Skins

Skins default document appearance may be modified using skins contained in sub-directories located at the following paths:

./_sisu/skin

~ /.sisu/skin

/etc/sisu/skin

more specifically, the following locations (or their /etc/sisu equivalent) should be used:

~ /.sisu/skin/doc

skins for individual documents;

~ /.sisu/skin/dir

skins for directories of matching names;

~ /.sisu/skin/site

site-wide skin modifying the site-wide appearance of documents.

Usually all skin files are placed in the document skin directory:

~ /.sisu/skin/doc

with softlinks being made to the skins contained there from other skin directories as required.

44. FAQ - Frequently Asked/Answered Questions

44.1 Why are urls produced with the -v (and -u) flag that point to a web server on port 8081?

Try the following rune:

- sisu -W

This should start the ruby webserver. It should be done after having produced some output as it scans the output directory for what to serve.

44.2 I cannot find my output, where is it?

The following should provide help on output paths:

- sisu –help env
- sisu -V [same as the previous command]
- sisu –help directory
- sisu –help path
- sisu -U [filename]
- man sisu

44.3 I do not get any pdf output, why?

SiSU produces LaTeX and pdflatex is run against that to generate pdf files.

If you use **Debian** the following will install the required dependencies

- aptitude install sisu-pdf

the following packages are required: tetex-bin, tetex-extra, latex-ucs

44.4 Where is the latex (or some other interim) output?

Try adding -M (for maintenance) to your command flags, e.g.:

- sisu -HpMv [filename]

this should result in the interim processing output being retained, and information being provided on where to find it.

- sisu –help directory
- sisu –help path

should also provide some relevant information as to where it is placed.

44.5 Why isn't SiSU markup XML

I worked with text and (though I find XML immensely valuable) disliked noise ... better to sidestep the question and say:

SiSU currently "understands" three XML input representations - or more accurately, converts from three forms of XML to native **SiSU** markup for processing. The three types correspond to SAX (structure described), DOM (structure embedded, whole document must be read before structure is correctly discernable) and node based (a tree) forms of XML document structure representation. Problem is I use them very seldom and check that all is as it should be with them seldom, so I would not be surprised if something breaks there, but as far as I know they are working. I will check and add an XML markup help page before the next release. There already is a bit of information in the man page under the title **SiSU** VERSION CONVERSION

sisu –to-sax [filename/wildcard]

sisu –to-dom [filename/wildcard]

sisu –to-node [filename/wildcard]

The XML should be well formed... must check, but lacks sensible headers. Suggestions welcome as to what to make of them. [For the present time I am satisfied that I can convert (both ways) between 3 forms of XML representation and **SiSU** markup].

sisu –from-xml2sst [filename/wildcard]

44.6 LaTeX claims to be a document preparation system for high-quality typesetting. Can the same be said about SiSU?

SiSU is not really about type-setting.

LaTeX is the ultimate computer instruction type-setting language for paper based publication.

LaTeX is able to control just about everything that happens on page and pixel, position letters kerning, space variation between characters, words, paragraphs etc. formula.

SiSU is not really about type-setting at all. It is about a lightweight markup instruction that provides enough information for an abstraction of the documents structure and objects, from which different forms of representation of the document can be generated.

SiSU with very little markup instruction is able to produce relatively high quality pdf by virtue of being able to generate usable default LaTeX; it produces "quality" html by generating the html directly; likewise it populates an SQL database in a useful way with the document in object sized chunks and its meta-data. But **SiSU** works on an abstraction of the document's structure and content and custom builds suitable uniform output. The html for browser viewing and pdf for paper viewing/publishing are rather different things with different needs for layout - as indeed is what is needed to store information in a database in searchable objects.

The pdfs or html produced for example by open office based on open document format and other office/word processor suits usually attempt to have similar looking outputs - your document rendered in html looks much the same, or in pdf... sisu is less this way, it seeks to

have a starting point with as little information about appearance as possible, and to come up with the best possible appearance for each output that can be derived based on this minimal information.

Where there are large document sets, it provides consistency in appearance in each output format for the documents.

The excuse for going this way is, it is a waste of time to think much about appearance when working on substantive content, it is the substantive content that is relevant, not the way it looks beyond the basic informational tags - and yet you want to be able to take advantage of as many useful different ways of representing documents as are available, and for various types of output to to be/look as good as it can for each medium/format in which it is presented, (with different mediums having different focuses) and **SiSU** tries to achieve this from minimal markup.

44.7 How do I create GIN or GiST index in Postgresql for use in SiSU

This at present needs to be done "manually" and it is probably necessary to alter the sample search form. The following is a helpful response from one of the contributors of GiN to Postgresql Oleg Bartunov 2006-12-06:

"I have tsearch2 slides which introduces tsearch2 http://www.sai.msu.su/~ megera/wiki/tsearch2slides

FTS in PostgreSQL is provided by tsearch2, which should works without any indices (GiST or GIN) ! Indices provide performance, not functionality.

In your example I'd do (simple way, just for demo):

0. compile, install tsearch2 and load tsearch2 into your database

    cd contrib/tsearch2; makemakeinstallmake installcheck; psql DB < tsearch2.sql

1. Add column fts, which holds tsvector

    alter table documents add column fts tsvector;

2. Fill fts column

    update document set fts = to_tsvector(clean);

3. create index - just for performance !

    create index fts_gin_idx on document using gin(fts);

4. Run vacuum

    vacuum analyze document;

That's all.

Now you can search:

select lid, metadata_tid, rank_cd(fts, q,2)as rank from document, plainto_tsquery('markup syntax') q where q @@ fts order by rank desc limit 10;

44.8 Where is version 1.0?

SiSU works pretty well as it is supposed to. Version 1.0 will have the current markup, and directory structure. At this point it is largely a matter of choice as to when the name change is made.

The feature set for html,[110] LaTeX/pdf and opendocument is in place. XML, and plaintext are in order.

html and LaTeX/pdf may be regarded as reference copy outputs

With regard to the populating of sql databases (postgresql and sqlite), there is a bit to be done.

We are still almost there.

[110] html w3c compliance has been largely met.

45. Editor Files, Syntax Highlighting

The directory:

./data/sisu/conf/editor-syntax-etc/

/usr/share/sisu/conf/editor-syntax-etc

contains rudimentary sisu syntax highlighting files for:

- (g)vim <http://www.vim.org>

  package: sisu-vim

status: largely done

there is a vim syntax highlighting and folds component

- gedit <http://www.gnome.org/projects/gedit>
- gobby <http://gobby.0x539.de/>

  file: sisu.lang

place in:

/usr/share/gtksourceview-1.0/language-specs

or

~ /.gnome2/gtksourceview-1.0/language-specs

status: very basic syntax highlighting

comments: this editor features display line wrap and is used by Goby!

- nano <http://www.nano-editor.org>

  file: nanorc

save as:

~ /.nanorc

status: basic syntax highlighting

comments: assumes dark background; no display line-wrap; does line breaks

- diakonos (an editor written in ruby) <http://purepistos.net/diakonos>

file: diakonos.conf

save as:

~ /.diakonos/diakonos.conf

includes:

status: basic syntax highlighting

comments: assumes dark background; no display line-wrap

- kate and kwrite <http://kate.kde.org>

file: sisu.xml

place in:

/usr/share/apps/katepart/syntax

or

~ /.kde/share/apps/katepart/syntax

[settings::configure kate::{highlighting,filetypes}]

[tools::highlighting::{markup,scripts}::**SiSU**]

- nedit <http://www.nedit.org>

  file: sisu_nedit.pats

  nedit -import sisu_nedit.pats

  status: a very clumsy first attempt [not really done]

  comments: this editor features display line wrap

- emacs <http://www.gnu.org/software/emacs/emacs.html>

  files: sisu-mode.el

  to file ~ /.emacs add the following 2 lines:

  (add-to-list 'load-path "/usr/share/sisu-examples/config/syntax_hi")

  (require 'sisu-mode.el)

  [not done / not yet included]

- vim and gvim <http://www.vim.org>

  files:

  package is the most comprehensive sisu syntax highlighting and editor environment provided to date (is for vim/ gvim, and is separate from the contents of this directory)

  status: this includes: syntax highlighting; vim folds; some error checking

  comments: this editor features display line wrap

NOTE:

[**SiSU** parses files with long lines or line breaks, but, display linewrap (without line-breaks) is a convenient editor feature to have for sisu markup]

46. Help Sources

For a summary of alternative ways to get help on **SiSU** try one of the following:

man page

man sisu_help

man2html

<file:///usr/share/doc/sisu/html/sisu_help.1.html>

<http://www.jus.uio.no/sisu/man/sisu_help.1.html>

sisu generated output - links to html

<file:///usr/share/doc/sisu/sisu_manual/sisu_help/index.html>

<http://sisudoc.org/sisu_manual/sisu_help/index.html>

<http://www.jus.uio.no/sisu/sisu_manual/sisu_help/index.html>

help sources lists

Alternative sources for this help sources page listed here:

man sisu_help_sources

<file:///usr/share/doc/sisu/sisu_manual/sisu_help_sources/index.html>

<file:///usr/share/doc/sisu/html/sisu_help_sources/index.html>

<http://sisudoc.org/sisu_manual/sisu_help_sources/index.html>

<http://www.jus.uio.no/sisu/sisu_manual/sisu_help_sources/index.html>

46.1 man pages

46.1.1 man

man sisu

man sisu_commands

man 7 sisu_complete

man sisu_configuration

man 8 sisu_faq

man sisu_filetypes

man sisu_help

man sisu_help_sources

man 8 sisu_howto

man sisu_introduction

man sisu_markup

man sisu_output_overview

man 7 sisu_pdf

man 7 sisu_postgresql

man 8 sisu_quickstart

man 8 sisu_remote

man 8 sisu_search

man sisu_skin

man 7 sisu_sqlite

man 8 sisu_syntax_highlighting

man 7 sisu_vim

man sisu_webrick

46.2 sisu generated output - links to html

Note **SiSU** documentation is prepared in **SiSU** and output is available in multiple formats including amongst others html, pdf, and odf which may be also be accessed via the html pages[111]

46.2.1 locally installed

<file:///usr/share/doc/sisu/sisu_manual/sisu_manual/index.html>

<file:///usr/share/doc/sisu/sisu_manual/sisu_help/index.html>

<file:///usr/share/doc/sisu/sisu_manual/sisu_help_sources/index.html>

/usr/share/doc/sisu/sisu_manual/sisu/index.html

/usr/share/doc/sisu/sisu_manual/sisu_commands/index.html

/usr/share/doc/sisu/sisu_manual/sisu_complete/index.html

/usr/share/doc/sisu/sisu_manual/sisu_configuration/index.html

/usr/share/doc/sisu/sisu_manual/sisu_description/index.html

/usr/share/doc/sisu/sisu_manual/sisu_examples/index.html

/usr/share/doc/sisu/sisu_manual/sisu_faq/index.html

/usr/share/doc/sisu/sisu_manual/sisu_filetypes/index.html

/usr/share/doc/sisu/sisu_manual/sisu_help/index.html

/usr/share/doc/sisu/sisu_manual/sisu_help_sources/index.html

/usr/share/doc/sisu/sisu_manual/sisu_howto/index.html

/usr/share/doc/sisu/sisu_manual/sisu_introduction/index.html
/usr/share/doc/sisu/sisu_manual/sisu_manual/index.html
/usr/share/doc/sisu/sisu_manual/sisu_markup/index.html
/usr/share/doc/sisu/sisu_manual/sisu_output_overview/index.html
/usr/share/doc/sisu/sisu_manual/sisu_pdf/index.html
/usr/share/doc/sisu/sisu_manual/sisu_postgresql/index.html
/usr/share/doc/sisu/sisu_manual/sisu_quickstart/index.html
/usr/share/doc/sisu/sisu_manual/sisu_remote/index.html
/usr/share/doc/sisu/sisu_manual/sisu_search/index.html
/usr/share/doc/sisu/sisu_manual/sisu_skin/index.html
/usr/share/doc/sisu/sisu_manual/sisu_sqlite/index.html
/usr/share/doc/sisu/sisu_manual/sisu_syntax_highlighting/index.html
/usr/share/doc/sisu/sisu_manual/sisu_vim/index.html
/usr/share/doc/sisu/sisu_manual/sisu_webrick/index.html

46.2.2 www.sisudoc.org

<http://sisudoc.org/sisu_manual/sisu_manual/index.html>

<http://sisudoc.org/sisu_manual/sisu/index.html>
<http://sisudoc.org/sisu_manual/sisu_commands/index.html>
<http://sisudoc.org/sisu_manual/sisu_complete/index.html>
<http://sisudoc.org/sisu_manual/sisu_configuration/index.html>
<http://sisudoc.org/sisu_manual/sisu_description/index.html>
<http://sisudoc.org/sisu_manual/sisu_examples/index.html>
<http://sisudoc.org/sisu_manual/sisu_faq/index.html>
<http://sisudoc.org/sisu_manual/sisu_filetypes/index.html>
<http://sisudoc.org/sisu_manual/sisu_help/index.html>
<http://sisudoc.org/sisu_manual/sisu_help_sources/index.html>
<http://sisudoc.org/sisu_manual/sisu_howto/index.html>
<http://sisudoc.org/sisu_manual/sisu_introduction/index.html>
<http://sisudoc.org/sisu_manual/sisu_manual/index.html>
<http://sisudoc.org/sisu_manual/sisu_markup/index.html>
<http://sisudoc.org/sisu_manual/sisu_output_overview/index.html>

<http://sisudoc.org/sisu_manual/sisu_pdf/index.html>

<http://sisudoc.org/sisu_manual/sisu_postgresql/index.html>

<http://sisudoc.org/sisu_manual/sisu_quickstart/index.html>

<http://sisudoc.org/sisu_manual/sisu_remote/index.html>

<http://sisudoc.org/sisu_manual/sisu_search/index.html>

<http://sisudoc.org/sisu_manual/sisu_skin/index.html>

<http://sisudoc.org/sisu_manual/sisu_sqlite/index.html>

<http://sisudoc.org/sisu_manual/sisu_syntax_highlighting/index.html>

<http://sisudoc.org/sisu_manual/sisu_vim/index.html>

<http://sisudoc.org/sisu_manual/sisu_webrick/index.html>

46.2.3 www.jus.uio.no/sisu

<http://www.jus.uio.no/sisu/sisu_manual/sisu_manual/index.html>

<http://www.jus.uio.no/sisu/sisu_manual/sisu/index.html>

<http://www.jus.uio.no/sisu/sisu_manual/sisu_commands/index.html>

<http://www.jus.uio.no/sisu/sisu_manual/sisu_complete/index.html>

<http://www.jus.uio.no/sisu/sisu_manual/sisu_configuration/index.html>

<http://www.jus.uio.no/sisu/sisu_manual/sisu_description/index.html>

<http://www.jus.uio.no/sisu/sisu_manual/sisu_examples/index.html>

<http://www.jus.uio.no/sisu/sisu_manual/sisu_faq/index.html>

<http://www.jus.uio.no/sisu/sisu_manual/sisu_filetypes/index.html>

<http://www.jus.uio.no/sisu/sisu_manual/sisu_help/index.html>

<http://www.jus.uio.no/sisu/sisu_manual/sisu_help_sources/index.html>

<http://www.jus.uio.no/sisu/sisu_manual/sisu_howto/index.html>

<http://www.jus.uio.no/sisu/sisu_manual/sisu_introduction/index.html>

<http://www.jus.uio.no/sisu/sisu_manual/sisu_manual/index.html>

<http://www.jus.uio.no/sisu/sisu_manual/sisu_markup/index.html>

<http://www.jus.uio.no/sisu/sisu_manual/sisu_output_overview/index.html>

<http://www.jus.uio.no/sisu/sisu_manual/sisu_pdf/index.html>

<http://www.jus.uio.no/sisu/sisu_manual/sisu_postgresql/index.html>

<http://www.jus.uio.no/sisu/sisu_manual/sisu_quickstart/index.html>

<http://www.jus.uio.no/sisu/sisu_manual/sisu_remote/index.html>

<http://www.jus.uio.no/sisu/sisu_manual/sisu_search/index.html>

<http://www.jus.uio.no/sisu/sisu_manual/sisu_skin/index.html>

<http://www.jus.uio.no/sisu/sisu_manual/sisu_sqlite/index.html>

<http://www.jus.uio.no/sisu/sisu_manual/sisu_syntax_highlighting/index.html>

<http://www.jus.uio.no/sisu/sisu_manual/sisu_vim/index.html>

<http://www.jus.uio.no/sisu/sisu_manual/sisu_webrick/index.html>

46.2.4 man2html

46.2.5 locally installed

<file:///usr/share/doc/sisu/html/sisu_manual.1.html>

<file:///usr/share/doc/sisu/html/sisu_help.1.html>

<file:///usr/share/doc/sisu/html/sisu_help_sources.1.html>

/usr/share/doc/sisu/html/sisu.1.html

/usr/share/doc/sisu/html/sisu_commands.1.html

/usr/share/doc/sisu/html/sisu_complete.7.html

/usr/share/doc/sisu/html/sisu_configuration.1.html

/usr/share/doc/sisu/html/sisu_faq.8.html

/usr/share/doc/sisu/html/sisu_help.1.html

/usr/share/doc/sisu/html/sisu_help_sources.1.html

/usr/share/doc/sisu/html/sisu_howto.8.html

/usr/share/doc/sisu/html/sisu_markup.1.html

/usr/share/doc/sisu/html/sisu_pdf.7.html

/usr/share/doc/sisu/html/sisu_postgresql.7.html

/usr/share/doc/sisu/html/sisu_quickstart.8.html

/usr/share/doc/sisu/html/sisu_remote.8.html

/usr/share/doc/sisu/html/sisu_search.8.html

/usr/share/doc/sisu/html/sisu_skin.1.html

/usr/share/doc/sisu/html/sisu_sqlite.7.html

/usr/share/doc/sisu/html/sisu_syntax_highlighting.8.html

/usr/share/doc/sisu/html/sisu_vim.7.html

/usr/share/doc/sisu/html/sisu_webrick.1.html

46.2.6 www.sisudoc.org

<http:///sisudoc.org/man/sisu_manual.1.html>

<http:///sisudoc.org/man/sisu_help.1.html>

<http:///sisudoc.org/man/sisu_help_sources.1.html>

<http://sisudoc.org/man/sisu.1.html>

<http://sisudoc.org/man/sisu_commands.1.html>

<http://sisudoc.org/man/sisu_complete.7.html>

<http://sisudoc.org/man/sisu_configuration.1.html>

<http://sisudoc.org/man/sisu_faq.8.html>

<http://sisudoc.org/man/sisu_help.1.html>

<http://sisudoc.org/man/sisu_help_sources.1.html>

<http://sisudoc.org/man/sisu_howto.8.html>

<http://sisudoc.org/man/sisu_markup.1.html>

<http://sisudoc.org/man/sisu_pdf.7.html>

<http://sisudoc.org/man/sisu_postgresql.7.html>

<http://sisudoc.org/man/sisu_quickstart.8.html>

<http://sisudoc.org/man/sisu_remote.8.html>

<http://sisudoc.org/man/sisu_search.8.html>

<http://sisudoc.org/man/sisu_skin.1.html>

<http://sisudoc.org/man/sisu_sqlite.7.html>

<http://sisudoc.org/man/sisu_syntax_highlighting.8.html>

<http://sisudoc.org/man/sisu_vim.7.html>

<http://sisudoc.org/man/sisu_webrick.1.html>

46.2.7 www.jus.uio.no/sisu

<http://www.jus.uio.no/sisu/man/sisu_manual.1.html>

<http://www.jus.uio.no/sisu/man/sisu_help.1.html>

<http://www.jus.uio.no/sisu/man/sisu_help_sources.1.html>

<http://www.jus.uio.no/sisu/man/sisu.1.html>

<http://www.jus.uio.no/sisu/man/sisu_commands.1.html>

<http://www.jus.uio.no/sisu/man/sisu_complete.7.html>

<http://www.jus.uio.no/sisu/man/sisu_configuration.1.html>

<http://www.jus.uio.no/sisu/man/sisu_faq.8.html> 2318

<http://www.jus.uio.no/sisu/man/sisu_help.1.html> 2319

<http://www.jus.uio.no/sisu/man/sisu_help_sources.1.html> 2320

<http://www.jus.uio.no/sisu/man/sisu_howto.8.html> 2321

<http://www.jus.uio.no/sisu/man/sisu_markup.1.html> 2322

<http://www.jus.uio.no/sisu/man/sisu_pdf.7.html> 2323

<http://www.jus.uio.no/sisu/man/sisu_postgresql.7.html> 2324

<http://www.jus.uio.no/sisu/man/sisu_quickstart.8.html> 2325

<http://www.jus.uio.no/sisu/man/sisu_remote.8.html> 2326

<http://www.jus.uio.no/sisu/man/sisu_search.8.html> 2327

<http://www.jus.uio.no/sisu/man/sisu_skin.1.html> 2328

<http://www.jus.uio.no/sisu/man/sisu_sqlite.7.html> 2329

<http://www.jus.uio.no/sisu/man/sisu_syntax_highlighting.8.html> 2330

<http://www.jus.uio.no/sisu/man/sisu_vim.7.html> 2331

<http://www.jus.uio.no/sisu/man/sisu_webrick.1.html> 2332

[11] named index.html or more extensively through sisu_manifest.html

DOCUMENT INFORMATION (METADATA)

Metadata

Document Manifest @

<http://www.jus.uio.no/sisu/sisu_manual/sisu_manual/sisu_manifest.html>

Dublin Core (DC)

DC tags included with this document are provided here.

DC Title: SiSU - Manual

DC Creator: Ralph Amissah

DC Rights: Copyright (C) Ralph Amissah 2007, part of SiSU documentation, License GPL 3

DC Type: information

DC Date created: 2002-08-28

DC Date issued: 2002-08-28

DC Date available: 2002-08-28

DC Date modified: 2007-08-30

DC Date: 2007-08-30

Version Information

Sourcefile: sisu_manual._sst

Filetype: SiSU text insert 0.58

Sourcefile Digest, MD5(sisu_manual._sst)= 978fcebd89bd099b135fc44be7e8073e

Skin_Digest: MD5(/home/ralph/grotto/theatre/dbld/builds/sisu/sisu/data/doc/sisu/sisu_markup_samples/sisu_m
20fc43cf3eb6590bc3399a1aef65c5a9

Generated

Document (metaverse) last generated: Tue Sep 25 02:51:58 +0100 2007

Generated by: SiSU 0.59.1 of 2007w39/2 (2007-09-25)

Ruby version: ruby 1.8.6 (2007-06-07 patchlevel 36) [i486-linux]

Information on this document copy and an unofficial List of Some web related information and sources

”Support Open Standards and Software Libre for the Information Technology Infrastructure” RA

Information on this document copy www.jus.uio.no/sisu/

Generated by SiSU found at www.jus.uio.no/sisu [SiSU 0.59.1 2007w39/2] www.sisudoc.org. SiSU is software for document structuring, publishing and search (using SiSU: object citation numbering, markup, meta-markup, and system) Copyright © 1997, current 2007 Ralph Amissah, All Rights Reserved.
SiSU is released under GPL 3 or later (www.fsf.org/licenses/gpl.html).

W3 since October 3 1993 SiSU 1997, current 2007.
SiSU presentations at **www.jus.uio.no/sisu/**

SiSU **pdf** versions can be found at:
http://www.jus.uio.no/sisu/sisu_manual/portrait.pdf
http://www.jus.uio.no/sisu/sisu_manual/landscape.pdf

SiSU **html** versions may be found at:
http://www.jus.uio.no/sisu/sisu_manual/toc.html or
http://www.jus.uio.no/sisu/sisu_manual/doc.html

SiSU Manifest of document output and metadata may be found at:
http://www.jus.uio.no/sisu/sisu_manual/sisu_manifest.html

SiSU found at: **www.jus.uio.no/sisu/**

Links that may be of interest at SiSU and elsewhere:

SiSU Manual
http://www.jus.uio.no/sisu/sisu_manual/

Book Samples and Markup Examples
http://www.jus.uio.no/sisu/SiSU/2.html

SiSU @ Wikipedia
http://en.wikipedia.org/wiki/SiSU

SiSU @ Freshmeat
http://freshmeat.net/projects/sisu/

SiSU @ Ruby Application Archive
http://raa.ruby-lang.org/project/sisu/

SiSU @ Debian
http://packages.qa.debian.org/s/sisu.html

SiSU Download

http://www.jus.uio.no/sisu/SiSU/download.html

SiSU Changelog

http://www.jus.uio.no/sisu/SiSU/changelog.html

SiSU help

http://www.jus.uio.no/sisu/sisu_manual/sisu_help/

SiSU help sources

http://www.jus.uio.no/sisu/sisu_manual/sisu_help_sources/

SiSU home:
www.jus.uio.no/sisu/