

SiSU - Manual

Ralph Amissah

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

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

Contents

**SiSU - Manual,
Ralph Amissah** **1**

What is SiSU? **1**

- 1. Introduction - What is SiSU? 1
- 2. How does sisu work? 4
- 3. Summary of features 4
- 4. Help 6
 - 4.1 SiSU Manual 6
 - 4.2 SiSU man pages 6
 - 4.3 SiSU built-in interactive help 7
- 5. Commands Summary 7
 - 5.1 Description 7
 - 5.2 Document Processing Command Flags 7
- 6. command line modifiers 13
- 7. database commands 13
- 8. Shortcuts, Shorthand for multiple flags 14
 - 8.1 Command Line with Flags - Batch Processing 14
- 9. Introduction to SiSU Markup 15
 - 9.1 Summary 15
 - 9.2 Markup Examples 15
 - 9.2.1 Online 15
 - 9.2.2 Installed 15
- 10. Markup of Headers 16
 - 10.1 Sample Header 16
 - 10.2 Available Headers 17
- 11. Markup of Substantive Text 18
 - 11.1 Heading Levels 18
 - 11.2 Font Attributes 19
 - 11.3 Indentation and bullets 19
 - 11.4 Footnotes / Endnotes 20

- 11.5 Links 21
 - 11.5.1 Naked URLs within text, dealing with urls 21
 - 11.5.2 Linking Text 21
 - 11.5.3 Linking Images 22
- 11.6 Grouped Text 23
 - 11.6.1 Tables 23
 - 11.6.2 Poem 24
 - 11.6.3 Group 25
 - 11.6.4 Code 26
- 11.7 Book index 28
- 12. Composite documents markup 29

Markup Syntax History **30**

- 13. Notes related to Files-types and Markup Syntax 30
- 14. SiSU filetypes 33
 - 14.1 .sst .ssm .ssi marked up plain text 33
 - 14.1.1 sisu text - regular files (.sst) 33
 - 14.1.2 sisu master files (.ssm) 33
 - 14.1.3 sisu insert files (.ssi) 33
 - 14.2 sisupod, zipped binary container (sisupod.zip, .ssp) 34
- 15. Experimental Alternative Input Representations 35
 - 15.1 Alternative XML 35
 - 15.1.1 XML SAX representation 35
 - 15.1.2 XML DOM representation 35
 - 15.1.3 XML Node representation 35
- 16. Configuration 36
 - 16.1 Determining the Current Configuration 36
 - 16.2 Configuration files (config.yml) 36
- 17. Skins 37
 - 17.1 Document Skin 37
 - 17.2 Directory Skin 37
 - 17.3 Site Skin 37

17.4 Sample Skins	37	<i>“Accelerando”</i> , Charles Stross	46
18. CSS - Cascading Style Sheets (for html, XHTML and XML)	38	<i>“Tainaron”</i> , Leena Krohn	46
19. Organising Content	38	<i>“Sphinx or Robot”</i> , Leena Krohn	46
19.1 Directory Structure and Mapping	38	<i>“War and Peace”</i> , Leo Tolstoy, PG Etext 2600	47
19.1.1 General Directories	38	<i>“Don Quixote”</i> , Miguel de Cervantes [Saavedra], translated by John Ormsby, PG Etext 996	47
19.1.2 Remote Directories	39	<i>“Gulliver’s Travels”</i> , Jonathan Swift, transcribed from the 1892 George Bell and Sons edition by David Price, PG Etext 829	48
19.1.3 Sisupod	39	<i>“Alice’s Adventures in Wonderland”</i> , Lewis Carroll, PG Etext 11	48
19.2 Organising Content	39	<i>“Through The Looking-Glass”</i> , Lewis Carroll, PG Etext 12	48
20. Homepages	40	<i>“Alice’s Adventures in Wonderland”</i> and <i>“Through The Looking-Glass”</i> , Lewis Carroll, PG Etexts 11 and 12	49
20.1 Home page and other custom built pages in a sub-directory	40	<i>“Gnu Public License 2”</i> , (GPL 2) Free Software Foundation	49
20.2 Home page within a skin	40	<i>“Gnu Public License v3 - Third discussion draft”</i> , (GPLv3) Free Software Foundation	49
21. Markup and Output Examples	41	<i>“Debian Social Contract”</i>	50
21.1 Markup examples	41	<i>“Debian Constitution v1.3”</i> , (simple/default markup)	50
21.2 A few book (and other) examples	41	<i>“Debian Constitution v1.3”</i> , (markup adjusted for output to more closely match the original)	50
21.2.1 <i>“Viral Spiral”</i> , David Bollier	41	<i>“Debian Constitution v1.2”</i> , (simple/default markup)	51
<i>“The Wealth of Networks”</i> , Yochai Benkler	42	<i>“Debian Constitution v1.2”</i> , (markup adjusted for output to more closely match the original)	51
<i>“Two Bits”</i> , Christopher Kelty	42	<i>“A Uniform Sales Terminology”</i> , Vikki Rogers and Albert Kritzer	51
<i>“Free Culture”</i> , Lawrence Lessig	42	<i>“The Autonomous Contract” 1997</i> - markup sample	52
<i>“CONTENT”</i> , Cory Doctorow	43		
<i>“Democratizing Innovation”</i> , by Eric von Hippel	43		
<i>“Free as in Freedom: Richard Stallman’s Crusade for Free Software”</i> , by Sam Williams	43		
<i>“Free For All: How Linux and the Free Software Movement Undercut the High Tech Titans”</i> , by Peter Wayner	44		
<i>“The Cathedral and the Bazaar”</i> , by Eric S. Raymond	44		
<i>“Down and out in the Magic Kingdom”</i> , Cory Doctorow	45		
<i>“Little Brother”</i> , Cory Doctorow	45		
<i>“For the Win”</i> , Cory Doctorow	45		

<i>“The Autonomous Contract Revisited”</i> - markup sample	52	27.1 Search - database frontend sample, utilising database and SiSU features, including object citation numbering (backend currently PostgreSQL)	62
<i>“United Nations Convention on Contracts for the International Sale of Goods”</i>	53	27.2 Search Form	62
/PECL/ the “Principles of European Contract Law”	53	28. Hyperrestraiier	63
21.3 SQL - PostgreSQL, SQLite	53	29. sisu_webrick	64
21.4 Lex Mercatoria as an example	54	29.1 Name	64
21.5 For good measure the markup for a document with lots of (simple) tables	54	29.2 Synopsis	64
21.6 And a link to the output of a reported case	54	29.3 Description	64
22. A Checklist of Output Features	55	29.4 Summary of man page	64
23. SiSU Search - Introduction	58	29.5 Document processing command flags	65
24. SQL	59	29.6 Further information	65
24.1 populating SQL type databases	59	29.7 Author	65
25. Postgresql	60	29.8 SEE ALSO	65
25.1 Name	60	30. Remote Source Documents	65
25.2 Description	60	Remote Document Output	66
25.3 Synopsis	60	31. Remote Output	66
25.4 Commands	60	31.1 commands	66
25.4.1 create and destroy database	60	31.2 configuration	66
25.4.2 import and remove documents	60	32. Remote Servers	67
26. Sqlite	61	Download information	68
26.1 Name	61	33. Download SiSU - Linux/Unix	68
26.2 Description	61	SiSU Current Version - Linux/Unix	68
26.3 Synopsis	61	Source (tarball tar.gz)	68
26.4 Commands	61	Git (source control management)	69
26.4.1 create and destroy database	61	Debian	69
26.4.2 import and remove documents	61	RPM	70
27. Introduction	62	Installation	72
		34. Installation	72
		34.1 Debian	72

34.2 Other Unix / Linux	73	38.1.1 Use of search functionality, an example using sqlite	82
34.2.1 source tarball	73	38.2 misc	84
35. SiSU Components, Dependencies and Notes	74	38.2.1 url for output files -u -U	84
35.1 sisu	74	38.2.2 toggle screen color	84
35.2 sisu-complete	75	38.2.3 verbose mode	84
35.3 sisu-examples	75	38.2.4 quiet mode	84
35.4 sisu-pdf	76	38.2.5 maintenance mode intermediate files kept -M	84
35.5 sisu-postgresql	76	38.2.6 start the webrick server	84
35.6 sisu-remote	76	38.3 remote placement of output	84
35.7 sisu-sqlite	76	39. Configuration Files	85
36. Quickstart - Getting Started Howto	77	40. Markup	85
36.1 Installation	77	40.1 Headers	85
36.1.1 Debian Installation	77	40.2 Font Face	86
36.1.2 RPM Installation	78	40.2.1 Bold	86
36.1.3 Installation from source	78	40.2.2 Italics	86
36.2 Testing SiSU, generating output	79	40.2.3 Underscore	86
36.2.1 basic text, plaintext, html, XML, ODF, EPUB	79	40.2.4 Strikethrough	86
36.2.2 LaTeX / pdf	79	40.3 Endnotes	86
36.2.3 relational database - postgresql, sqlite	79	40.4 Links	87
36.3 Getting Help	80	40.5 Number Titles	87
36.3.1 The man pages	80	40.6 Line operations	87
36.3.2 Built in help	80	40.7 Tables	87
36.3.3 The home page	80	40.8 Grouped Text	88
36.4 Markup Samples	80	40.9 Composite Document	88
HowTo	81	41. Change Appearance	89
37. Getting Help	81	41.1 Skins	89
37.1 SiSU “man” pages	81	41.2 CSS	89
37.2 SiSU built-in help	82	Extracts from the README	90
37.3 Command Line with Flags - Batch Processing	82	42. README	90
38. Setup, initialisation	82	42.1 Online Information, places to look	91
38.1 initialise output directory	82		

42.2 Installation	91	44.4 Where is the latex (or some other interim) out- put?	104
42.2.1 Debian	91	44.5 Why isn't SiSU markup XML	104
42.2.2 RPM	92	44.6 LaTeX claims to be a document preparation system for high-quality typesetting. Can the same be said about SiSU?	104
42.2.3 Source package .tgz	92	44.7 Can the SiSU markup be used to prepare for a LaTeX automatic building of an index to the work?	105
42.2.4 to use setup.rb	92	44.8 Can the conversion from SiSU to LaTeX be modified if we have special needs for the La- TeX, or do we need to modify the LaTeX manually?	105
42.2.5 to use install (praped with "Rake")	92	44.9 How do I create GIN or GiST index in Post- gresql for use in SiSU	105
42.2.6 to use install (praped with "Rant")	93	44.10 Are there some examples of using Ferret Search with a SiSU repository?	106
42.3 Dependencies	93	Have you had any reports of building SiSU from tar on Mac OS 10.4?	106
42.4 Quick start	96	44.12 Where is version 1?	106
42.5 Configuration files	97	44.13 What is the difference between version 1 and 2?	107
42.6 Use General Overview	97	45. Who might be interested in the SiSU feature set?	107
42.7 Help	97	46. Work Needed	108
42.8 Directory Structure	98	47. Wishlist	108
42.9 Configuration File	98	48. Editor Files, Syntax Highlighting	109
42.10 Markup	98	49. Help Sources	111
42.11 Additional Things	98	49.1 man pages	111
42.12 License	98	49.2 sisu generated output - links to html	111
42.13 SiSU Standard	99	49.2.1 www.sisudoc.org	111
Extracts from man 8 sisu	100	49.3 man2html	111
43. Post Installation Setup	100	49.3.1 locally installed	111
43.1 Post Installation Setup - Quick start	100	49.3.2 www.jus.uio.no/sisu	111
43.2 Document markup directory	101		
43.2.1 Configuration files	101		
43.2.2 Debian INSTALLATION Note	101		
43.2.3 Document Resource Configuration	102		
43.2.4 Skins	102		
44. FAQ - Frequently Asked/Answered Questions	103		
44.1 Why are urls produced with the -v (and -u) flag that point to a web server on port 8081 ?	103		
44.2 I cannot find my output, where is it?	103		
44.3 I do not get any pdf output, why?	103		

Metadata	113
SiSU Metadata, document information	113

1	SiSU - Manual, Ralph Amissah
2	What is SiSU?

1. Introduction - What is SiSU? 3

SiSU is a framework for document structuring, publishing (in multiple open standard formats) 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 (including the population of sql databases) that (can) share a common numbering system for the citation of text within a document. 4

SiSU is developed under an open source, software libre license (GPL3). Its use case for development is work with medium to large document sets and cope with evolving document formats/ representation technologies. Documents are prepared once, and generated as need be to update the technical presentation or add additional output formats. Various output formats (including search related output) share a common mechanism for cross-output-format citation. 5

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, EPUB, 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. 6

Source document preparation and output generation is a two step 7

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 (to take advantage of the strengths of different ways of representing documents) various standard open output formats including plain text, HTML, XHTML, XML, EPUB, OpenDocument, LaTeX or PDF files, and populate an SQL database with objects¹ (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.

8 In preparing a **SiSU** document you optionally provide semantic information related to the 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 builds² the different forms of output requested.

9 **SiSU** works with an abstraction of the document based on its

¹objects include: headings, paragraphs, verse, tables, images, but not footnotes/endnotes which are numbered separately and tied to the object from which they are referenced.

²i.e. the html, pdf, epub, 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.

structure which is comprised of its headings³ and objects⁴, 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 basis for citing material within a document across the different output format types. This is significant as page numbers are not well suited to the digital age, in web publishing, changing a browser’s default font or using a different browser can mean that text will appear on a different page; and publishing in different formats, html, landscape and portrait pdf etc. again page numbers are not useful to cite text. Dealing with documents at an object level together with object numbering also has implications for search that **SiSU** is able to take advantage of.

One of the challenges of maintaining documents is to keep them 10 in a format that allows use of them independently of proprietary platforms. Consider issues related to 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 producing documents in multiple non-proprietary open formats including html, pdf⁵ ODF,⁶ and EPUB.⁷ 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

³the different heading levels

⁴units of text, primarily paragraphs and headings, also any tables, poems, code-blocks

⁵Specification submitted by Adobe to ISO to become a full open ISO specification

<<http://www.linux-watch.com/news/NS7542722606.html>>

⁶ISO standard ISO/IEC 26300:2006

⁷An open standard format for e-books

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 when it became an ISO standard for office applications and the archival of documents), EPUB was introduced in 2009; and allows the technical representations 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).

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

11 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.

12 **SiSU** also provides other features such as concordance files and document content certificates, 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 thesauri, together with the flexibility of **SiSU** offers great possibilities.

13 **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** .

14 <<http://www.sisudoc.org/>>

16 2. How does sisu work?

17 **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

- 19 Γ 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.
- 20 Γ 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].
- 21 Γ 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, strikethrough 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, epub, xml, odf, latex, pdf, sql)
- 22 Γ 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:
 - 23 Γ html - both as a single scrollable text and a segmented document
 - 24 Γ xhtml
 - 25 Γ epub

- 26 Γ XML - both in sax and dom style xml structures for further development as required
- 27 Γ ODF - open document format, the iso standard for document storage
- 28 Γ LaTeX - used to generate pdf
- 29 Γ pdf (via LaTeX)
- 30 Γ sql - population of an sql database, (at the same object level that is used to cite text within a document)
- 31 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))
- 32 Γ 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, EPUB, xml, sqlite, postgresql), this numbering system can be used to reference content.
- 33 Γ 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 hyperstraier].
- Γ 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, epub in 2009 and in future html5 output sometime in future, without modification of existing prepared texts 34
- Γ SQL search aside, documents are generated as required and static once generated. 35
- Γ 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) 36
- Γ document source (plaintext utf-8) if shared on the net may be used as input and processed locally to produce the different document outputs 37
- Γ 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 38
- Γ generated document outputs may automatically be posted to remote sites. 39
- Γ for basic document generation, the only software dependency is **Ruby** , and a few standard Unix tools (this covers plaintext, 40

HTML, EPUB, 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.

41 Γ as a developers tool it is flexible and extensible
 42 Syntax highlighting for **SiSU** markup is available for a number of
 text editors.
 43 **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 representa-
 tions of it which may be rather different from each other and used
 for different purposes, whether layout and publishing, or search of
 content
 44 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)..
 45 the solution arrived at is by extracting structural information about
 the document (about headings within the document) and by track-
 ing 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 46

4.1 SiSU Manual 47

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

`<http://sisudoc.org/sisu/sisu_manual/>` 49

The manual can be generated from source, found respectively, either
 within the **SiSU** tarball or installed locally at:

`./data/doc/sisu/markup-samples/sisu_manual` 51

`/usr/share/doc/sisu/markup-samples/sisu_manual` 52

move to the respective directory and type e.g.: 53

`sisu sisu_manual.ssm` 54

4.2 SiSU man pages 55

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

`man sisu` 57

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/markup-samples/sisu_manual` 59

Once installed, directory equivalent to: 60

`/usr/share/doc/sisu/markup-samples/sisu_manual` 61

Available man pages are converted back to html using
 man2html: 62

`/usr/share/doc/sisu/html/` 63

`./data/doc/sisu/html` 64

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

66 Γ <[various sisu man pages](#)> ⁸

67 Γ <[sisu.1](#)> ⁹

68 4.3 SiSU built-in interactive help

69 This is particularly useful for getting the current sisu
setup/environment information:

70 `sisu -help`

71 `sisu -help [subject]`

72 `sisu -help commands`

73 `sisu -help markup`

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

75 `sisu -V` [environment information, same as above command]

76 `sisu` (on its own provides version and some help information)

77 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).

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

⁸<<http://www.jus.uio.no/sisu/man/>>

⁹<<http://www.jus.uio.no/sisu/man/sisu.1.html>>

5. Commands Summary 79

5.1 Description 80

SiSU **SiSU** is a document publishing system, that from a simple 81
single marked-up document, produces multiple of output formats
including: plaintext, html, xhtml, XML, epub, odt (odf text), La-
TeX, pdf, 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](http://www.jus.uio.no/sisu)>

5.2 Document Processing Command Flags 82

-a [filename/wildcard] 83

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] 84

see `-xhtml`

-color-toggle [filename/wildcard] 85

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). Alias `-c`

-concordance [filename/wildcard] 86

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`). Alias `-w`

- 87 **-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.
- 88 **-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.
- 89 **-c** [**filename/wildcard**]
 see **-color-toggle**
- 90 **-dal** [**filename/wildcard/url**]
 assumed for most other flags, creates new intermediate files for processing (document abstraction) that is used in all subsequent processing of other output. This step is assumed for most processing flags. To skip it see **-n**. Alias **-m**
- 91 **-delete** [**filename/wildcard**]
 see **-zap**
- 92 **-D** [**instruction**] [**filename**]
 see **-pg**
- 93 **-d** [**-db-[database type (sqlite|pg)]**] [**-[instruction]**] [**filename**]
 see **-sqlite**
- 94 **-epub** [**filename/wildcard**]
 produces an epub document, [sisu version 2 only] (filename.epub). Alias **-e**
- e** [**filename/wildcard**] 95
 see **-epub**
- F** [**-webserv=webrick**] 96
 see **-sample-search-form**
- git** [**filename/wildcard**] 97
 produces or updates markup source file structure in a git repo (experimental and subject to change). Alias **-g**
- g** [**filename/wildcard**] 98
 see **-git**
- harvest *.ss[tm]** 99
 makes two lists of sisu output based on the sisu markup documents in a directory: list of author and authors works (year and titles), and; list by topic with titles and author. Makes use of header metadata fields (author, title, date, topic_register). Can be used with maintenance (**-M**) and remote placement (**-R**) flags.
- help** [**topic**] 100
 provides help on the selected topic, where topics (keywords) include: list, (com)mands, short(cuts), (mod)ifiers, (env)ironment, markup, syntax, headers, headings, endnotes, tables, example, customise, skin, (dir)ectories, path, (lang)uage, db, install, setup, (conf)igure, convert, termsheet, search, sql, features, license
- html** [**filename/wildcard**] 101
 produces html output, segmented text with table of contents (toc.html and index.html) and the document in a single file (scroll.html). Alias **-h**
- h** [**filename/wildcard**] 102
 see **-html**
- I** [**filename/wildcard**] 103

	see <code>-texinfo</code>		
104	-i [filename/wildcard]		
	see <code>-manpage</code>		
105	-L		
	prints license information.		
106	-machine [filename/wildcard/url]		
	see <code>-dal</code> (document abstraction level/layer)		
107	-maintenance [filename/wildcard/url]		
	maintenance mode files created for processing preserved and their locations indicated. (also see <code>-V</code>). Alias <code>-M</code>		
108	-manpage [filename/wildcard]		
	produces man page of file, not suitable for all outputs. Alias <code>-i</code>		
109	-M [filename/wildcard/url]		
	see <code>-maintenance</code>		
110	-m [filename/wildcard/url]		
	see <code>-dal</code> (document abstraction level/layer)		
111	-no-ocn		
	[with <code>-html</code> <code>-pdf</code> or <code>-epub</code>] switches off object citation numbering. Produce output without identifying numbers in margins of html or LaTeX/pdf output.		
112	-N [filename/wildcard/url]		
	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). <code>-NV</code> for verbose digest output to screen.		
113	-n [filename/wildcard/url]		
	skip the creation of intermediate processing files (document abstraction) if they already exist, this skips the equivalent of <code>-m</code> which is otherwise assumed by most processing flags.		
	-odf [filename/wildcard/url]		114
	see <code>-odt</code>		
	-odt [filename/wildcard/url]		115
	output basic document in opendocument file format (opendocument.odt). Alias <code>-o</code>		
	-o [filename/wildcard/url]		116
	see <code>-odt</code>		
	-pdf [filename/wildcard]		117
	produces LaTeX pdf (portrait.pdf & landscape.pdf). Default paper size is set in config file, or document header, or provided with additional command line parameter, e.g. <code>-papersize-a4</code> preset sizes include: 'A4', U.S. 'letter' and 'legal' and book sizes 'A5' and 'B5' (system defaults to A4). Alias <code>-p</code>		
	-pg [instruction] [filename]		118
	database postgresql (<code>-pgsql</code> may be used instead) possible instructions, include: <code>-createdb</code> ; <code>-create</code> ; <code>-dropall</code> ; <code>-import [filename]</code> ; <code>-update [filename]</code> ; <code>-remove [filename]</code> ; see database section below. Alias <code>-D</code>		
	-po [language_directory/filename language_directory]		119
	see <code>-po4a</code>		
	-po4a [language_directory/filename language_directory]		120
	produces .pot and po files for the file in the languages specified by the language directory. SiSU markup is placed in subdirectories named with the language code, e.g. <code>en/ fr/ es/</code> . The sisu config file must set the output directory structure to multilingual. <code>v3</code> , experimental		
	-P [language_directory/filename language_directory]		121
	see <code>-po4a</code>		
	-p [filename/wildcard]		122
	see <code>-pdf</code>		

- quiet [filename/wildcard]**
quiet less output to screen.
- 124 **-q [filename/wildcard]**
see `-quiet`
- 125 **-rsync [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 `-scp`. Alias `-R`
- 126 **-R [filename/wildcard]**
see `-rsync`
- 127 **-r [filename/wildcard]**
see `-scp`
- 128 **-sample-search-form [-webserv=webrick]**
generate examples of (naive) cgi search form for sqlite and postgresql depends on your already having used sisu to populate an sqlite and/or postgresql 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 hyperrestraier for sisu. Alias `-F`
- 129 **-scp [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 `-rsync`. Alias `-r`
- sqlite -[instruction] [filename]** 130
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. Alias `-d`
- sisupod** 131
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). Alias `-S`
- sisupod [filename/wildcard]** 132
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 be-

- comes more space efficient to zip the sisu markup directory, rather than the individual files for sending). See the `-S` option without `[filename/wildcard]`. Alias `-S`
- 133 `–source [filename/wildcard]`
copies sisu markup file to output directory. Alias `-s`
- 134 `-S`
see `–sisupod`
- 135 `-S [filename/wildcard]`
see `–sisupod`
- 136 `-s [filename/wildcard]`
see `–source`
- 137 `–texinfo [filename/wildcard]`
produces texinfo and info file, (view with `pinfo`). Alias `-I`
- 138 `–txt [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). Alias `-t`
- 139 `-T [filename/wildcard (*.termsheet.rb)]`
standard form document builder, preprocessing feature
- 140 `-t [filename/wildcard]`
see `–txt`
- 141 `–urls [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`. Alias `-U`
- `-U [filename/wildcard]` 142
see `–urls`
- `-u [filename/wildcard]` 143
provides url mapping of output files for the flags requested for processing, also see `-U`
- `–v2 [filename/wildcard]` 144
invokes the sisu v2 document parser/generator. This is the default and is normally omitted.
- `–v3 [filename/wildcard]` 145
invokes the sisu v3 document parser/generator. Currently under development and incomplete, v3 requires `≥ ruby1.9.2p180`. You may run `sisu3` instead.
- `–verbose [filename/wildcard]` 146
provides verbose output of what is being generated, where output is placed (and error messages if any), as with `-u` flag provides a url mapping of files created for each of the processing flag requests. Alias `-v`
- `-V` 147
on its own, provides **SiSU** version and environment information (`sisu –help env`)
- `-V [filename/wildcard]` 148
even more verbose than the `-v` flag.
- `-v` 149
on its own, provides **SiSU** version information
- `-v [filename/wildcard]` 150
see `–verbose`
- `–webrick` 151
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]. Alias -W

152 **-W**

see -webrick

153 **-wordmap [filename/wildcard]**

see -concordance

154 **-w [filename/wildcard]**

see -concordance

155 **-xhtml [filename/wildcard]**

produces xhtml/XML output for browser viewing (sax parsing).
Alias -b

156 **-xml-dom [filename/wildcard]**

produces XML output with deep document structure, in the nature of dom. Alias -X

157 **-xml-sax [filename/wildcard]**

produces XML output shallow structure (sax parsing). Alias -x

158 **-X [filename/wildcard]**

see -xml-dom

159 **-x [filename/wildcard]**

see -xml-sax

160 **-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])

161 **-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.

-zap [filename/wildcard]

162

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. Alias -Z

-Z [filename/wildcard]

163

see -zap

164 **6. command line modifiers**165 **–no-ocn**

[with `–html` `–pdf` or `–epub`] switches off object citation numbering. Produce output without identifying numbers in margins of html or LaTeX/pdf output.

166 **–no-annotate**

strips output text of editor endnotes^{*1} denoted by asterisk or dagger/plus sign

167 **–no-asterisk**

strips output text of editor endnotes^{*2} denoted by asterisk sign

168 **–no-dagger**

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

^{*1}square brackets

^{*2}square brackets

⁺¹square brackets

7. database commands 169

`dbi` - database interface 170

`-D` or `–pgsql` set for postgresql `-d` or `–sqlite` default set for sqlite `-d` is modifiable with `–db=[database type (pgsql or sqlite)]` 171

–pg -v –createall 172

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 alphanumeric and underscores.

–pg -v –import 173

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

–pg -v –update 174

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

–pg –remove 175

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

–pg –dropall 176

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

The `-v` is for verbose output. 177

178 **8. Shortcuts, Shorthand for multiple flags**179 **-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.

180 **-0 to -5 [filename or wildcard]**

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

181 **-0**

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

182 **-1**

-mhewpy

183 **-2**

-mhewpaoy

184 **-3**

-mhewpAobxXyY

185 **-4**

-mhewpAobxXDyY -import

186 **-5**

-mhewpAobxXDyY -update

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

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

189 **8.1 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. 190

Running sisu (alone without any flags, filenames or wildcards) 191 brings up the interactive help, as does any sisu command that is not recognised. Enter to escape.

192	9. Introduction to SiSU Markup ¹⁰	For brief descriptive summary of markup history	208
193	9.1 Summary	<code>sisu -query-history</code>	209
194	SiSU source documents are plaintext (UTF-8) ¹¹ files	or if for a particular version:	210
195	All paragraphs are separated by an empty line.	<code>sisu -query-0.38</code>	211
196	Markup is comprised of:		
197	Γ 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)		
198	Γ 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:		
199	Γ heading levels defines document structure		
200	Γ text basic attributes, italics, bold etc.		
201	Γ grouped text (objects), which are to be treated differently, such as code blocks or poems.		
202	Γ footnotes/endnotes		
203	Γ linked text and images		
204	Γ paragraph actions, such as indent, bulleted, numbered-lists, etc.		
205	Some interactive help on markup is available, by typing <code>sisu</code> and selecting <code>markup</code> or <code>sisu -help markup</code>		
206	To check the markup in a file:		
207	<code>sisu -identify [filename].sst</code>		
		9.2 Markup Examples	212
		9.2.1 Online	213
		Online markup examples are available together with the respective outputs produced from <code><http://www.jus.uio.no/sisu/SiSU/examples.html></code> or from <code><http://www.jus.uio.no/sisu/sisu_examples/></code>	214
		There is of course this document, which provides a cursory overview of <code>sisu</code> markup and the respective output produced: <code><http://www.jus.uio.no/sisu/sisu_markup/></code>	215
		an alternative presentation of markup syntax: <code>/usr/share/doc/sisu/on_markup.txt.gz</code>	216
		9.2.2 Installed	217
		With SiSU installed sample skins may be found in: <code>/usr/share/doc/sisu/markup-samples</code> (or equivalent directory) and if <code>sisu-markup-samples</code> is installed also under: <code>/usr/share/doc/sisu/markup-samples-non-free</code>	218

¹⁰From sometime after SiSU 0.58 it should be possible to describe SiSU markup using SiSU, which though not an original design goal is useful.

¹¹files should be prepared using UTF-8 character encoding

219 10. Markup of Headers

220 Headers contain either: semantic meta-data about a document,
which can be used by any output module of the program, or; pro-
cessing instructions.

221 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 char-
acter in a line of text) followed by a space and the comment:

222 % this would be a comment

223 10.1 Sample Header

224 This current document is loaded by a master document that has a
header similar to this one:

225 % SiSU master 2.0

```
@title: SiSU
:subtitle: Manual
```

```
@creator: :author: Amissah, Ralph
```

```
@rights: Copyright (C) Ralph Amissah 2007, part of SiSU
documentation, License GPL 3
```

```
@classify:
:type: information
:topic_register: SiSU>manual;electronic
documents:SiSU>manual
:subject: ebook, epublising, electronic book,
electronic publishing,
electronic document, electronic citation, data
structure,
citation systems, search
```

```
% used_by: manual
```

```
@date:
:published: 2008-05-22
:created: 2002-08-28
:issued: 2002-08-28
:available: 2002-08-28
:modified: 2010-03-03
```

```
@make:
:num_top: 1
:breaks: new=C; break=1
:skin: skin_sisu_manual
:bold: /Gnu|Debian|Ruby|SiSU/
:manpage: name=sisu - documents: markup, structuring,
publishing in multiple standard formats, and search;
synopsis=sisu
[-abcDdeFhIiMmNnopqRrSsTtUuVvwXxYyZz0-9] [filename/wildcard
]
. sisu [-Ddcv] [instruction]
. sisu [-CcFLSVvW]
. sisu --v2 [operations]
. sisu --v3 [operations]
```

```
@links:
{ SiSU Homepage }http://www.sisudoc.org/
{ SiSU Manual }http://www.sisudoc.org/sisu/sisu_manual/
{ Book Samples & Markup Examples
}http://www.jus.uio.no/sisu/SiSU/examples.html
{ SiSU Download
}http://www.jus.uio.no/sisu/SiSU/download.html
{ SiSU Changelog
}http://www.jus.uio.no/sisu/SiSU/changelog.html
{ SiSU Git repo
}http://git.sisudoc.org/?p=code/sisu.git;a=summary
{ SiSU List Archives
}http://lists.sisudoc.org/pipermail/sisu/
{ SiSU @ Debian }http://packages.qa.debian.org/s/sisu.html
```

```

{ SiSU Project @ Debian
}http://qa.debian.org/developer.php?login=sisu@lists.sisudoc.org
{ SiSU @ Wikipedia }http://en.wikipedia.org/wiki/SiSU

```

226 10.2 Available Headers

227 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 the form @headername: or on the next line and indented by once space :subheadername: All Dublin Core meta tags are available

228 **@identifier:** information or instructions

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

230 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.

231 This is a sample header

```
232 % SiSU 2.0 [declared file-type identifier with markup
version]
```

```
233 @title: [title text] [this header is the only one that is
mandatory]
```

```
    :subtitle: [subtitle if any]
    :language: English
```

234 @creator:

```
    :author: [Lastname, First names]
    :illustrator: [Lastname, First names]
    :translator: [Lastname, First names]
    :prepared_by: [Lastname, First names]
```

235 @date:

```
    :published: [year or yyyy-mm-dd]
```

```

:created: [year or yyyy-mm-dd]
:issued: [year or yyyy-mm-dd]
:available: [year or yyyy-mm-dd]
:modified: [year or yyyy-mm-dd]
:valid: [year or yyyy-mm-dd]
:added_to_site: [year or yyyy-mm-dd]
:translated: [year or yyyy-mm-dd]

```

@rights:

```

:copyright: Copyright (C) [Year and Holder]
:license: [Use License granted]
:text: [Year and Holder]
:translation: [Name, Year]
:illustrations: [Name, Year]

```

236

@classify:

```

:topic_register: SiSU:markup
sample:book;book:novel:fantasy
:type:
:subject:
:description:
:keywords:
:abstract:
:isbn: [ISBN]
:loc: [Library of Congress classification]
:dewey: [Dewey classification]
:pg: [Project Gutenberg text number]

```

237

```
@links: { SiSU }http://www.sisudoc.org
        { FSF }http://www.fsf.org
```

238

@make:

```

:skin: skin_name [skins change default settings related
to the appearance of documents generated]
:num_top: 1
:headings: [text to match for each level
(e.g. PART; Chapter; Section; Article; or another:
none; BOOK|FIRST|SECOND; none; CHAPTER;)]
:breaks: new=:C; break=1
:promo: sisu, ruby, sisu.search.libre, open.society
:bold: [regular expression of words/phrases to be made

```

239

```

bold]
  :italics: [regular expression of words/phrases to
italicise]
240  @original:
      :language: [language]
241  @notes:
      :comment:
      :prefix: [prefix is placed just after table of
contents]

```

11. Markup of Substantive Text 242

11.1 Heading Levels 243

Heading levels are :A~ ,:B~ ,:C~ ,1~ ,2~ ,3~ ... :A - :C being part / 244
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 245
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 246
divider]

:C~ [heading text] Third level heading [this is a heading level 247
divider]

1~ [heading text] Top level heading preceding substantive text of 248
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 249
of document or sub-heading 3 , the heading level that would nor-
mally 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 250
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, 251

% 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)

252 **11.2 Font Attributes**

253 **markup example:**

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

normal text

`{emphasis}`* [note: can be configured to be represented by bold, italics or underscore]

`!{bold text}!`

`/italics/`

`_{underscore}_`

`"{citation}"`

`^ {superscript}^`

`{subscript}`,

`+{inserted text}+`

`-{strikethrough}-`

`#{monospace}#`

255 **resulting output:**

256 normal text, **emphasis** , **bold text** , *italics*, underline, “ci-

tation”, ^{superscript}, _{subscript}, inserted text, ~~strikethrough~~,
monospace

normal text 257

emphasis [note: can be configured to be represented by bold, italics or underscore] 258

bold text 259

italics 260

underline 261

“citation” 262

^{superscript} 263

_{subscript} 264

inserted text 265

~~strikethrough~~ 266

monospace 267

11.3 Indentation and bullets 268

markup example: 269

ordinary paragraph 270

.1 indent paragraph one step

.2 indent paragraph two steps

.9 indent paragraph nine steps

resulting output: 271

ordinary paragraph 272

indent paragraph one step 273

indent paragraph two steps 274

275		indent para- graph nine steps	12		290
				markup example:	291
276	markup example:			normal text [~] { self contained endnote marker & endnote in one } [~] continues	292
277	_* bullet text			resulting output:	293
	_1* bullet text, first indent			normal text ¹³ continues	294
	_2* bullet text, two step indent			markup example:	295
278	resulting output:			normal text [~] {* unnumbered asterisk footnote/endnote, insert multiple asterisks if required } [~] continues	296
279	Γ bullet text				
280	Γ bullet text, first indent			normal text [~] {** another unnumbered asterisk footnote/endnote } [~] continues	
281	Γ bullet text, two step indent			resulting output:	297
282	Numbered List (not to be confused with headings/titles, (document structure))			normal text * continues	298
				normal text ** continues	299
283	markup example:			markup example:	300
284	# numbered list	numbered list 1., 2., 3, etc.		normal text [~] [* editors notes, numbered asterisk footnote/endnote series] [~] continues	301
	_# numbered list	numbered list indented a., b., c., d., etc.		normal text [~] [+ editors notes, numbered asterisk footnote/endnote series] [~] continues	
285	11.4 Footnotes / Endnotes			resulting output:	302
				normal text ^{*3} continues	303
286	Footnotes and endnotes are marked up at the location where they would be indicated within a text. They are automatically num- bered. The output type determines whether footnotes or endnotes will be produced			normal text ⁺² continues	304
287	markup example:			Alternative endnote pair notation for footnotes/endnotes:	305
288	[~] { a footnote or endnote } [~]			¹² a footnote or endnote	
289	resulting output:			¹³ self contained endnote marker & 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	

		320	<code>deb http://www.jus.uio.no/sisu/archive unstable main non-free</code>	
306	% note the endnote marker " <code>^^</code> "		where a code block is used there is neither decoration nor hyper-	321
	normal text <code>^^</code> continues		linking, code blocks are discussed later in this document	
	<code>^^</code> endnote text following the paragraph in which the		resulting output:	322
	marker occurs		<code>deb http://www.jus.uio.no/sisu/archive unstable main</code>	323
307	the standard and pair notation cannot be mixed in the same docu-		<code>non-free</code>	
	ment		<code>deb-src http://www.jus.uio.no/sisu/archive unstable main</code>	
			<code>non-free</code>	
308	11.5 Links			
309	11.5.1 Naked URLs within text, dealing with urls		11.5.2 Linking Text	324
310	urls found within text are marked up automatically. A url within		To link text or an image to a url the markup is as follows	325
	text is automatically hyperlinked to itself and by default decorated		markup example:	326
	with angled braces, unless they are contained within a code block		about { SiSU } <code>http://url.org</code> markup	327
	(in which case they are passed as normal text), or escaped by a pre-		resulting output:	328
	ceding underscore (in which case the decoration is omitted).		about <code><SiSU></code> markup	329
311	markup example:		A shortcut notation is available so the url link may also be provided	330
312	normal text <code>http://www.sisudoc.org/</code> continues		automatically as a footnote	
313	resulting output:		markup example:	331
314	normal text <code><http://www.sisudoc.org/></code> continues		about { <code>^^</code> SiSU } <code>http://url.org</code> markup	332
315	An escaped url without decoration		resulting output:	333
316	markup example:		about <code><SiSU></code> ¹⁴ markup	334
317	normal text <code>_http://www.sisudoc.org/</code> continues		Internal document links to a tagged location, including an	335
			ocn	
	<code>deb http://www.jus.uio.no/sisu/archive unstable main</code>		markup example:	336
	<code>non-free</code>		about { text links } <code>#link_text</code>	337
318	resulting output:		resulting output:	338
319	normal text <code>http://www.sisudoc.org/</code> continues		about <code><text links></code>	339

¹⁴`<http://www.sisudoc.org/>`

340 Shared document collection link

341 **markup example:**

```
342 about { SiSU book markup examples
343 }:SiSU/examples.html
```

343 **resulting output:**

344 about <SiSU book markup examples>



351

345 11.5.3 Linking Images

346 **markup example:**

```
347 { tux.png 64x80 }image
```

```
% various url linked images
```

```
{tux.png 64x80 "a better way" }http://www.sisudoc.org/
{GnuDebianLinuxRubyBetterWay.png 100x101 "Way Better -
with Gnu/Linux, Debian and Ruby" }http://www.sisudoc.org/
{^^ ruby_logo.png "Ruby" }http://www.ruby-lang.org/en/
```

348 **resulting output:**



349



350



15

352

353 linked url footnote shortcut

```
{^^ [text to link] }http://url.org
% maps to: { [text to link] }http://url.org ~{ http://url.org
}~
354
```

% which produces hyper-linked text within a document/paragraph, with an endnote providing the url for the text location used in the hyperlink

text marker

355

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. 356

11.6 Grouped Text

11.6.1 Tables

359 Tables may be prepared in two either of two forms

360 **markup example:**

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

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
```

362 **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

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

365 **markup example:** ¹⁶

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

```
{table~h 24; 12; 12; 12; 12; 12; 12;}
```

¹⁵<http://www.ruby-lang.org/en/>

¹⁶Table from the Wealth of Networks by Yochai Benkler
http://www.jus.uio.no/sisu/the_wealth_of_networks.yochai_benkler

357				Jan. 2001	Jan. 2002	Jan. 2003	Jan. 2004	July 2004	June 2006
358	Contributors*			10	472	2,188	9,653		
	Active contributors**			9	212	846	3,228	8,442	
	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

371

372 **basic markup:**
 373 `poem{`
 Your poem here
`}poem`

Each verse in a poem is given an object number.

374 **markup example:**
 375 `poem{`

```

    '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.'"

```

`}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

```

376

377

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.'"

378 11.6.3 Group

379 basic markup:

380 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

381

382

```

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

```

```
}group
```

383 **resulting output:**

384

```

'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

385

Code tags `{ ... }` (used as with other group tags described above) 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

386

tags. They are however used in the same way as group or poem tags.

387 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]

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

389 ‘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."
```

From **SiSU** 2.7.7 on you can number codeblocks by placing a hash 390
after the opening code tag `code{#` as demonstrated here:

```
1 |                   ‘Fury said to a                   391
2 |                   mouse, That he
3 |                   met in the
4 |                   house,
5 | "Let us
6 |                   both go to
7 |                   law: I will
8 |                   prosecute
9 |                   YOU. --Come,
10 |                   I'll take no
11 |                   denial; We
12 |                   must have a
13 |                   trial: For
14 |                   really this
15 |                   morning I've
16 |                   nothing
17 |                   to do."
18 |                   Said the
19 |                   mouse to the
20 |                   cur, "Such
```

```

21 |         a trial,
22 |           dear Sir,
23 |             With
24 |               no jury
25 |                 or judge,
26 |                   would be
27 |     wasting
28 | our
29 | breath."
30 |     "I'll be
31 |       judge, I'll
32 |         be jury,"
33 |           Said
34 |             cunning
35 |               old Fury:
36 |                 "I'll
37 |                   try the
38 |                     whole
39 |                       cause,
40 |                         and
41 |                           condemn
42 |                             you
43 |                               to
44 |                                 death.'"

```

392 11.7 Book index

393 To make an index append to paragraph the book index term relates to it, using an equal sign and curly braces.

394 Currently two levels are provided, a main term and if needed a sub-term. Sub-terms are separated from the main term by a colon.

```

395     Paragraph containing main term and sub-term.
        ={Main term:sub-term}

```

396 The index syntax starts on a new line, but there should not be an

empty line between paragraph and index markup.

The structure of the resulting index would be:

```

Main term, 1
sub-term, 1

```

Several terms may relate to a paragraph, they are separated by a semicolon. If the term refers to more than one paragraph, indicate the number of paragraphs.

```

Paragraph containing main term, second term and
sub-term.

```

```
={first term; second term: sub-term}
```

The structure of the resulting index would be:

```

First term, 1,
Second term, 1,
sub-term, 1

```

If multiple sub-terms appear under one paragraph, they are separated under the main term heading from each other by a pipe symbol.

```

Paragraph containing main term, second term and
sub-term.

```

```
={Main term:sub-term+1|second sub-term}
```

```

A paragraph that continues discussion of the first
sub-term

```

The plus one in the example provided indicates the first sub-term spans one additional paragraph. The logical structure of the resulting index would be:

```

Main term, 1,
sub-term, 1-3,
second sub-term, 1,

```

407 **12. Composite documents markup**

408 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**.

409 basic markup for importing a document into a master document

410 `<< filename1.sst`

`<< filename2.ssi`

411 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.

412 `<< filename.ssi`

`<<{filename.ssi}`

`% using textlink alternatives`

`<< |filename.ssi|@|^ |`

413	Markup Syntax History	
	13. Notes related to Files-types and Markup Syntax	414
	2.0 introduced new headers and is therefore incompatible with 1.0 though otherwise the same with the addition of a couple of tags (i.e. a superset)	415
	0.38 is substantially current for version 1.0	416
	depreciated 0.16 supported, though file names were changed at 0.37	417
	<code>Γ sisu -query=[sisu version [0.38] or 'history]</code>	418
	provides a short history of changes to SiSU markup	419
	SiSU 2.0 (2010-03-06:09/6) same as 1.0, apart from the changing of headers and the addition of a monospace tag related headers now grouped, e.g.	420
	<code>@title:</code>	421
	<code>:subtitle:</code>	
	<code>@creator:</code>	
	<code>:author:</code>	
	<code>:translator:</code>	
	<code>:illustrator:</code>	
	<code>@rights:</code>	
	<code>:text:</code>	
	<code>:illustrations:</code>	
	see document markup samples, and <code>sisu -help</code> headers	422
	the monospace tag takes the form of a hash <code>'#'</code>	423
	<code>#{ this enclosed text would be monospaced }#</code>	424
	1.0 (2009-12-19:50/6) same as 0.69	425
	0.69 (2008-09-16:37/2) (same as 1.0) and as previous (0.57) with the addition of book index tags	426
	<code>/^ ={.+?}\$ /</code>	427

- e.g. appended to a paragraph, on a new-line (without a blank line in between) logical structure produced assuming this is the first text “object”
- 429 ={GNU/Linux community
distribution:Debian+2|Fedora|Gentoo;Free Software
Foundation+5}
- 430 Free Software Foundation, 1-6
 GNU/Linux community distribution, 1
 Debian, 1-3
 Fedora, 1
 Gentoo,
- 431 **0.66** (2008-02-24:07/7) same as previous, adds semantic tags, [experimental and not-used]
- 432 /[[:;]]{.+?}[[:;]] [a-z+]/
- 433 **0.57** (2007w34/4) **SiSU** 0.57 is the same as 0.42 with the introduction of some a shortcut to use the headers @title and @creator in the first heading [expanded using the contents of the headers @title: and @author:]
- 434 :A~ @title by @author
- 435 **0.52** (2007w14/6) declared document type identifier at start of text/document:
- 436 **SiSU** 0.52
- 437 or, backward compatible using the comment marker:
- 438 % **SiSU** 0.38
- 439 variations include ‘**SiSU** (text|master|insert) [version]’ and ‘sisu-[version]’
- 440 **0.51** (2007w13/6) skins changed (simplified), markup unchanged
- 441 **0.42** (2006w27/4) * (asterisk) type endnotes, used e.g. in relation to author
- 442 **SiSU** 0.42 is the same as 0.38 with the introduction of some additional endnote types,
- 443 Introduces some variations on endnotes, in particular the use of the asterisk
- 444 ~{* for example for describing an author }~ and ~{** for describing a second author }~
- 445 * for example for describing an author
- 446 ** for describing a second author
- 447 and
- 448 ~[* my note]~ or ~[+ another note]~
- 449 which numerically increments an asterisk and plus respectively
- 450 *1 my note +1 another note
- 451 **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)
- 452 **SiSU** 0.38 introduced alternative experimental header and heading/structure markers,
- 453 @headername: and headers :A~ :B~ :C~ 1~ 2~ 3~
- 454 as the equivalent of:
- 455 0~headername and headers 1~ 2~ 3~ 4~ 5~ 6~
- 456 The internal document markup of **SiSU** 0.16 remains valid and standard Though note that **SiSU** 0.37 introduced a new file naming convention
- 457 **SiSU** has in effect two sets of levels to be considered, using 0.38 notation A-C headings/levels, pre-ordinary paragraphs /pre-substantive text, and 1-3 headings/levels, levels which are followed by ordinary text. This may be conceptualised as levels A,B,C, 1,2,3, and using such letter number notation, in effect: A must exist, optional B and C may follow in sequence (not strict) 1 must

exist, optional 2 and 3 may follow in sequence i.e. there are two independent heading level sequences A,B,C and 1,2,3 (using the 0.16 standard notation 1,2,3 and 4,5,6) on the positive side: the 0.38 A,B,C,1,2,3 alternative makes explicit an aspect of structuring documents in **SiSU** that is not otherwise obvious to the newcomer (though it appears more complicated, is more in your face and likely to be understood fairly quickly); the substantive text follows levels 1,2,3 and it is ‘nice’ to do most work in those levels

458 **0.37** (2006w09/7) introduced new file naming convention, .sst
(text), .ssm (master), .ssi (insert), markup syntax unchanged

459 **SiSU** 0.37 introduced new file naming convention, using the file
extensions .sst .ssm and .ssi to replace .s1 .s2 .s3 .r1 .r2 .r3 and
.si

460 this is captured by the following file ‘rename’ instruction:

```
461 rename 's/_s[123]$/_sst/' *.s{1,2,3}
rename 's/_r[123]$/_ssm/' *.r{1,2,3}
rename 's/_si$/_ssi/' *.si
```

462 The internal document markup remains unchanged, from **SiSU**
0.16

463 **0.35** (2005w52/3) sisupod, zipped content file introduced

464 **0.23** (2005w36/2) utf-8 for markup file

465 **0.22** (2005w35/3) image dimensions may be omitted if rmagick is
available to be relied upon

466 **0.20.4** (2005w33/4) header 0~links

467 **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

468 **SiSU** 0.16 (0.15 development branch) introduced the use of

469 the header 0~ and headings/structure 1~ 2~ 3~ 4~ 5~ 6~

470 in place of the 0.1 header, heading/structure notation

SiSU 0.1 headers and headings structure represented by header 471
0{~ and headings/structure 1{ 2{ 3{ 4{~ 5{ 6{

472 14. SiSU filetypes

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

474 14.1 .sst .ssm .ssi marked up plain text

475 **SiSU** documents are prepared as plain-text (utf-8) files with **SiSU**
markup. They may make reference to and contain images (for ex-
ample), which are stored in the directory beneath them `.sisu/image`.
SiSU plaintext markup files are of three types that may be dis-
tinguished by the file extension used: regular text `.sst`; master doc-
uments, 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.

476 **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.

477 **SiSU** source markup can be shared with the command:

478 `sisu -s [filename]`

479 14.1.1 `sisu` text - regular files (`.sst`)

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

481 [<http://www.sisudoc.org/sisu/sisu_markup>](http://www.sisudoc.org/sisu/sisu_markup)

482 [<http://www.sisudoc.org/sisu/sisu_manual>](http://www.sisudoc.org/sisu/sisu_manual)

483 14.1.2 `sisu` master files (`.ssm`)

484 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 485
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 486
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 docu-
ments.

Note: a secondary file of the composite document is built prior to 487
processing with the same prefix and the suffix `.sst` ¹⁷

[<http://www.sisudoc.org/sisu/sisu_markup>](http://www.sisudoc.org/sisu/sisu_markup) 488

[<http://www.sisudoc.org/sisu/sisu_manual>](http://www.sisudoc.org/sisu/sisu_manual) 489

490 14.1.3 `sisu` insert files (`.ssi`)

491 Inserts are documents prepared solely for the purpose of being in-
corporated into one or more master documents. They resemble
regular **SiSU** text files except they are ignored by the **SiSU** pro-
cessor. 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)

492 <http://www.sisudoc.org/sisu/sisu_manual>

502

493 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)

494 **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.

495 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.

496 The command to create a sisupod is:

497 `sisu -S [filename]`

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

499 `sisu -S`

500 **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.

501 <http://www.sisudoc.org/sisu/sisu_commands>

¹⁷.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.

503 **15. Experimental Alternative Input Representations**

519

or the same:

504 **15.1 Alternative XML**

505 **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.

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

507 `sisu -to-sax [filename/wildcard] or sisu -to-sxs [filename/wildcard]`

508 `sisu -to-dom [filename/wildcard] or sisu -to-sxd [filename/wildcard]`

509 `sisu -to-node [filename/wildcard] or sisu -to-sxn [filename/wildcard]`

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

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

512 or the same:

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

514 **15.1.1 XML SAX representation**

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

516 `sisu -to-sax [filename/wildcard] or sisu -to-sxs [filename/wildcard]`

517 To convert from any sisu xml representation back to sst

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

521 **15.1.2 XML DOM representation**

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

522 `sisu -to-dom [filename/wildcard] or sisu -to-sxd [filename/wildcard]` 523

To convert from any sisu xml representation back to sst

524 `sisu -from-xml2sst [filename/wildcard [.sxs.xml,.sxd.xml,sxn.xml]]` 525

or the same:

526 `sisu -from-sxml [filename/wildcard [.sxs.xml,.sxd.xml,sxn.xml]]` 527

528 **15.1.3 XML Node representation**

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

529 `sisu -to-node [filename/wildcard] or sisu -to-sxn [filename/wildcard]` 530

To convert from any sisu xml representation back to sst

531 `sisu -from-xml2sst [filename/wildcard [.sxs.xml,.sxd.xml,sxn.xml]]` 532

or the same:

533 `sisu -from-sxml [filename/wildcard [.sxs.xml,.sxd.xml,sxn.xml]]` 534

535 16. Configuration

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

536 16.1 Determining the Current Configuration

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

```
538     sisu -v
```

539 which is an alias for:

```
540     sisu -help env
```

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

If **SiSU** is installed a sample `sisurc.yml` may be found in `/etc/-sisu/sisurc.yml` 552

542 16.2 Configuration files (config.yml)

543 **SiSU** configuration 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.

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

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

```
546     ./_sisu/sisurc.yml
```

```
547     ~/.sisu/sisurc.yml
```

```
548     /etc/sisu/sisurc.yml
```

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

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

553 **17. Skins**

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

555 ./_sisu/skin
 556 ~/_sisu/skin
 557 /etc/sisu/skin

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

559 ./skin/doc
 560 ./skin/dir
 561 ./skin/site

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

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

564 **17.1 Document Skin**

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

566 @skin: skin_united_nations

567 **17.2 Directory Skin**

568 A directory may be mapped on to a particular skin, so all documents
 569 within that directory take on a particular appearance. If a skin
 570 exists in the skin/dir with the same name as the document directory,
 571 it will automatically be used for each of the documents in that

572 directory, (except where a document specifies the use of another
 573 skin, in the skin/doc directory).

574 A personal habit is to place all skins within the doc directory, and
 575 symbolic links as needed from the site, or dir directories as re-
 576 quired.

570 **17.3 Site Skin**

571 A site skin, modifies the program default skin.

572 **17.4 Sample Skins**

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

574 /etc/sisu/skin/doc and /usr/share/doc/sisu/markup-
 575 samples/samples/_sisu/skin/doc

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

578 /usr/share/doc/sisu/markup-samples-non-free/samples/_-
 579 sisu/skin/doc

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

582 /usr/share/doc/sisu/markup-samples-non-free/samples/_-
 583 sisu/skin/yml (or equivalent directory)

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

580 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`.

581 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.

582 HTML: `html.css`

583 XML DOM: `dom.css`

584 XML SAX: `sax.css`

585 XHTML: `xhtml.css`

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

587 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.¹⁸

¹⁸SiSU 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 588

19.1 Directory Structure and Mapping 589

590 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`)

591 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.

19.1.1 General Directories 592

593 `./subject_name/`

```

% files stored at this level e.g.  sisu_manual.sst
./subject_name/_sisu
% configuration file e.g.  sisurc.yml
./subject_name/_sisu/skin
% skins in various skin directories doc, dir, site, yml
./subject_name/_sisu/css
./subject_name/_sisu/image
% images for documents contained in this directory
./subject_name/_sisu/mm

```

594 19.1.2 Remote Directories

```

595  ./subject_name/
% containing sub_directories named after the generated
files from which they are made
./subject_name/src
% contains shared source files text and binary e.g.
sisu_manual.sst and sisu_manual.sst.zip
./subject_name/_sisu
% configuration file e.g.  sisurc.yml
./subject_name/_sisu/skin

```

19.1.3 Sisupod

```

./sisupod/
% files stored at this level e.g.  sisu_manual.sst
./sisupod/_sisu
% configuration file e.g.  sisurc.yml
./sisupod/_sisu/skin
% skins in various skin directories doc, dir, site, yml
./sisupod/_sisu/css
./sisupod/_sisu/image
% images for documents contained in this directory
./sisupod/_sisu/mm

```

19.2 Organising Content

599 20. Homepages

600 **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:

- 601 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)
- 602 2. through providing what you want as the home page in a skin,
- 603 Document sets are contained in directories, usually organised by site or subject. Each directory can/should have its own home page. See the section on directory structure and organisation of content.

604 20.1 Home page and other custom built pages in a sub-directory

605 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:

```
606     sisu -CC
```

607 20.2 Home page within a skin

608 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

```

611 **21. Markup and Output Examples**612 **21.1 Markup examples**

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

614 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.

21.2 A few book (and other) examples

615



19

616

21.2.1 “Viral Spiral”, David Bollier

617

<”Viral Spiral”, David Bollier>

618

¹⁹Reproduced 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

<document manifest> ²⁰
 < html, segmented text>
 < html, scroll, document in one>
 < epub>
 < pdf, landscape>
 < pdf, portrait>
 < odf:odt, open document text>
 < xhtml scroll>
 < xml, sax>
 < xml, dom>
 < plain text utf-8>
 < concordance>
 < dcc, document content certificate (digests)>
 <markup source text>
 <markup source (zipped) pod>

619 **“The Wealth of Networks”, Yochai Benkler**

620

<”The Wealth of Networks”, Yochai Benkler>
 <document manifest> ²¹
 < html, segmented text>
 < html, scroll, document in one>
 < epub>
 < pdf, landscape>
 < pdf, portrait>
 < odf:odt, open document text>
 < xhtml scroll>
 < xml, sax>
 < xml, dom>

²⁰<http:

[//www.sisudoc.org/sisu/viral_spiral.david_bollier/sisu_manifest.html](http://www.sisudoc.org/sisu/viral_spiral.david_bollier/sisu_manifest.html)>

²¹<http://www.sisudoc.org/sisu/the_wealth_of_networks.yochai_benkler/sisu_manifest.html>

< plain text utf-8>
 < concordance>
 < dcc, document content certificate (digests)>
 <markup source text>
 <markup source (zipped) pod>

“Two Bits”, Christopher Kelty

621

622

<”Two Bits”, Christopher Kelty>
 <document manifest> ²²
 < html, segmented text>
 < html, scroll, document in one>
 < epub>
 < pdf, landscape>
 < pdf, portrait>
 < odf:odt, open document text>
 < xhtml scroll>
 < xml, sax>
 < xml, dom>
 < plain text utf-8>
 < concordance>
 < dcc, document content certificate (digests)>
 <markup source text>
 <markup source (zipped) pod>

“Free Culture”, Lawrence Lessig

623

624

<”Free Culture”, Lawrence Lessig>

²²<http:

[//www.sisudoc.org/sisu/two_bits.christopher_kelty/sisu_manifest.html](http://www.sisudoc.org/sisu/two_bits.christopher_kelty/sisu_manifest.html)>

<document manifest> ²³
 < html, segmented text>
 < html, scroll, document in one>
 < epub>
 < pdf, landscape>
 < pdf, portrait>
 < odf:odt, open document text>
 < xhtml scroll>
 < xml, sax>
 < xml, dom>
 < plain text utf-8>
 < concordance>
 < dcc, document content certificate (digests)>
 <markup source text>
 <markup source (zipped) pod>

625 **“CONTENT”, Cory Doctorow**

626 <”CONTENT”, Cory Doctorow>
 <document manifest> ²⁴
 < html, segmented text>
 < html, scroll, document in one>
 < epub>
 < pdf, landscape>
 < pdf, portrait>
 < odf:odt, open document text>
 < xhtml scroll>
 < xml, sax>
 < xml, dom>
 < plain text utf-8>

²³ <http:

[//www.sisudoc.org/sisu/free_culture.lawrence_lessig/sisu_manifest.html](http://www.sisudoc.org/sisu/free_culture.lawrence_lessig/sisu_manifest.html)

²⁴ <http://www.sisudoc.org/sisu/content.cory_doctorow/sisu_manifest.html>

< concordance>
 < dcc, document content certificate (digests)>
 <markup source text>
 <markup source (zipped) pod>

“Democratizing Innovation”, by Eric von Hippel

627

628

<”Democratizing Innovation”, by Eric von Hippel>
 <document manifest> ²⁵
 < html, segmented text>
 < html, scroll, document in one>
 < epub>
 < pdf, landscape>
 < pdf, portrait>
 < odf:odt, open document text>
 < xhtml scroll>
 < xml, sax>
 < xml, dom>
 < plain text utf-8>
 < concordance>
 < dcc, document content certificate (digests)>
 <markup source text>
 <markup source (zipped) pod>

“Free as in Freedom: Richard Stallman’s Crusade for Free Software”, by Sam Williams

629

630

<”Free as in Freedom: Richard Stallman’s Crusade for Free Software”, by Sam Williams>

²⁵ <http://www.sisudoc.org/sisu/democratizing_innovation.eric_von_hippel/sisu_manifest.html>

<document manifest> ²⁶
 < html, segmented text>
 < html, scroll, document in one>
 < epub>
 < pdf, landscape>
 < pdf, portrait>
 < odf:odt, open document text>
 < xhtml scroll>
 < xml, sax>
 < xml, dom>
 < plain text utf-8>
 < concordance>
 < dcc, document content certificate (digests)>
 <markup source text>
 <markup source (zipped) pod>

< xhtml scroll>
 < xml, sax>
 < xml, dom>
 < plain text utf-8>
 < concordance>
 < dcc, document content certificate (digests)>
 <markup source text>
 <markup source (zipped) pod>

“The Cathedral and the Bazaar”, by Eric S. Raymond

633

<“The Cathedral and the Bazaar”, by Eric S. Raymond>

634

<document manifest> ²⁸
 < html, segmented text>
 < html, scroll, document in one>
 < epub>
 < pdf, landscape>
 < pdf, portrait>
 < odf:odt, open document text>
 < xhtml scroll>
 < xml, sax>
 < xml, dom>
 < plain text utf-8>
 < concordance>
 < dcc, document content certificate (digests)>
 <markup source text>
 <markup source (zipped) pod>

631 **“Free For All: How Linux and the Free Software Movement Undercut the High Tech Titans”, by Peter Wayner**

632 <“Free For All: How Linux and the Free Software Movement Undercut the High Tech Titans”, by Peter Wayner>

<document manifest> ²⁷
 < html, segmented text>
 < html, scroll, document in one>
 < epub>
 < pdf, landscape>
 < pdf, portrait>
 < odf:odt, open document text>

²⁶<http://www.sisudoc.org/sisu/free_as_in_freedom.richard_stallman_crusade_for_free_software.sam_williams/sisu_manifest.html>

²⁷<http://www.sisudoc.org/sisu/free_for_all.peter_wayner/sisu_manifest.html>

“Down and out in the Magic Kingdom”, Cory Doctorow

636

```
<"Down and out in the Magic Kingdom", Cory Doctorow>
<document manifest> 29
< html, segmented text>
< html, scroll, document in one>
< epub>
< pdf, landscape>
< pdf, portrait>
< odf:odt, open document text>
< xhtml scroll>
< xml, sax>
< xml, dom>
< plain text utf-8>
< concordance>
< dcc, document content certificate (digests)>
<markup source text>
<markup source (zipped) pod>
```

637

“Little Brother”, Cory Doctorow

638

```
<"Little Brother", Cory Doctorow>
<document manifest> 30
< html, segmented text>
< html, scroll, document in one>
< epub>
```

²⁸http://www.sisudoc.org/sisu/the_cathedral_and_the_bazaar.eric_s_raymond/sisu_manifest.html

²⁹http://www.sisudoc.org/sisu/down_and_out_in_the_magic_kingdom.cory_doctorow/sisu_manifest.html

³⁰http://www.sisudoc.org/sisu/little_brother.cory_doctorow/sisu_manifest.html

635

```
< pdf, landscape>
< pdf, portrait>
< odf:odt, open document text>
< xhtml scroll>
< xml, sax>
< xml, dom>
< plain text utf-8>
< concordance>
< dcc, document content certificate (digests)>
<markup source text>
<markup source (zipped) pod>
```

“For the Win”, Cory Doctorow

639

640

```
<"For the Win", Cory Doctorow>
<document manifest> 31
< html, segmented text>
< html, scroll, document in one>
< epub>
< pdf, landscape>
< pdf, portrait>
< odf:odt, open document text>
< xhtml scroll>
< xml, sax>
< xml, dom>
< plain text utf-8>
< concordance>
< dcc, document content certificate (digests)>
<markup source text>
<markup source (zipped) pod>
```

³¹http://www.sisudoc.org/sisu/for_the_win.cory_doctorow/sisu_manifest.html

641 **“Accelerando”, Charles Stross**

642

```

<"Accelerando", Charles Stross>
  <document manifest> 32
  < html, segmented text>
  < html, scroll, document in one>
  < epub>
  < pdf, landscape>
  < pdf, portrait>
  < odf:odt, open document text>
  < xhtml scroll>
  < xml, sax>
  < xml, dom>
  < plain text utf-8>
  < concordance>
  < dcc, document content certificate (digests)>
  <markup source text>
  <markup source (zipped) pod>

```

643 **“Tainaron”, Leena Krohn**

644

```

<"Tainaron", Leena Krohn>
  <document manifest> 33
  < html, segmented text>
  < html, scroll, document in one>
  < epub>
  < pdf, landscape>
  < pdf, portrait>

```

³²<http:[//www.sisudoc.org/sisu/accelerando.charles_stross/sisu_manifest.html](http://www.sisudoc.org/sisu/accelerando.charles_stross/sisu_manifest.html)³³<http:[//www.sisudoc.org/sisu/tainaron.leena_krohn.1998/sisu_manifest.html](http://www.sisudoc.org/sisu/tainaron.leena_krohn.1998/sisu_manifest.html)

```

  < odf:odt, open document text>
  < xhtml scroll>
  < xml, sax>
  < xml, dom>
  < plain text utf-8>
  < concordance>
  < dcc, document content certificate (digests)>
  <markup source text>
  <markup source (zipped) pod>

```

“Sphinx or Robot”, Leena Krohn

645



646

647

```

<"Sphinx or Robot", Leena Krohn>
  <document manifest> 34
  < html, segmented text>
  < html, scroll, document in one>
  < epub>
  < pdf, landscape>
  < pdf, portrait>
  < odf:odt, open document text>
  < xhtml scroll>
  < xml, sax>
  < xml, dom>
  < plain text utf-8>
  < concordance>
  < dcc, document content certificate (digests)>

```

³⁴<http://www.sisudoc.org/sisu/sphinx_or_robot.leena_krohn.1996/sisu_manifest.html>

<markup source text>
 <markup source (zipped) pod>

648 **“War and Peace”, Leo Tolstoy, PG Etext 2600**

649 <“War and Peace”, Leo Tolstoy> ³⁵
 <document manifest> ³⁶
 < html, segmented text>
 < html, scroll, document in one>
 < epub>
 < pdf, landscape>
 < pdf, portrait>
 < odf:odt, open document text>

³⁵<http:

[//www.jus.uio.no/sisu/war_and_peace.leo_tolstoy/toc.html](http://www.jus.uio.no/sisu/war_and_peace.leo_tolstoy/toc.html)> [U+258C] 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 [U+3014]bΩSiSUΩb[U+3015], 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 [U+3014]uΩvery little markup
 requiredΩu[U+3015] 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 [U+3014]lt[U+3015]:insert1[U+3014]gt[U+3015]
 and [U+3014]lt[U+3015]:insert2[U+3014]gt[U+3015] and the legal notices are
 available for similar insertion into the next Project Gutenberg text prepared
 for [U+3014]bΩSiSUΩb[U+3015], should there be one. [U+258C] I did a
 stylesheet/skin for the Gutenberg Project, ... I may have to remove.

³⁶<http:

[//www.sisudoc.org/sisu/war_and_peace.leo_tolstoy/sisu_manifest.html](http://www.sisudoc.org/sisu/war_and_peace.leo_tolstoy/sisu_manifest.html)>

< xhtml scroll>
 < xml, sax>
 < xml, dom>
 < plain text utf-8>
 < concordance>
 < dcc, document content certificate (digests)>
 <markup source text>
 <markup source (zipped) pod>

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

650

<“Don Quixote”, Miguel de Cervantes [Saavedra]>

<document manifest> ³⁷
 < html, segmented text>
 < html, scroll, document in one>
 < epub>
 < pdf, landscape>
 < pdf, portrait>
 < odf:odt, open document text>
 < xhtml scroll>
 < xml, sax>
 < xml, dom>
 < plain text utf-8>
 < concordance>
 < dcc, document content certificate (digests)>
 <markup source text>
 <markup source (zipped) pod>

651

“Gulliver’s Travels”, Jonathan Swift, transcribed from the 1892 George Bell and Sons edition by David Price, PG Etext 829

653

<”Gulliver’s Travels”, Jonathan Swift>
 <document manifest> ³⁸
 < html, segmented text>
 < html, scroll, document in one>
 < epub>
 < pdf, landscape>
 < pdf, portrait>
 < odf:odt, open document text>
 < xhtml scroll>
 < xml, sax>
 < xml, dom>
 < plain text utf-8>
 < concordance>
 < dcc, document content certificate (digests)>
 <markup source text>
 <markup source (zipped) pod>

“Alice’s Adventures in Wonderland”, Lewis Carroll, PG Etext 11

655

<”Alice’s Adventures in Wonderland”, Lewis Carroll>
 <document manifest> ³⁹
 < html, segmented text>

³⁷<http://www.sisudoc.org/sisu/don_quixote.miguel_de_cervantes/sisu_manifest.html>
³⁸<http://www.sisudoc.org/sisu/gullivers_travels.jonathan_swift/sisu_manifest.html>
³⁹<http://www.sisudoc.org/sisu/alices_adventures_in_wonderland.lewis_carroll/sisu_manifest.html>

< html, scroll, document in one>
 < epub>
 < pdf, landscape>
 < pdf, portrait>
 < odf:odt, open document text>
 < xhtml scroll>
 < xml, sax>
 < xml, dom>
 < plain text utf-8>
 < concordance>
 < dcc, document content certificate (digests)>
 <markup source text>
 <markup source (zipped) pod>

“Through The Looking-Glass”, Lewis Carroll, PG Etext 12

656

<”Through The Looking-Glass”, Lewis Carroll>
 <document manifest> ⁴⁰
 < html, segmented text>
 < html, scroll, document in one>
 < epub>
 < pdf, landscape>
 < pdf, portrait>
 < odf:odt, open document text>
 < xhtml scroll>
 < xml, sax>
 < xml, dom>
 < plain text utf-8>
 < concordance>
 < dcc, document content certificate (digests)>

657

⁴⁰<http://www.sisudoc.org/sisu/through_the_looking_glass.lewis_carroll/sisu_manifest.html>

<markup source text>
 <markup source (zipped) pod>

658 **“Alice’s Adventures in Wonderland” and “Through The
 Looking-Glass”, Lewis Carroll, PG Etexts 11 and
 12**

659 <“Alice’s Adventures in Wonderland” and “Through The Looking-Glass”, Lewis
 Carroll>

<document manifest> ⁴¹
 < html, segmented text>
 < html, scroll, document in one>
 < epub>
 < pdf, landscape>
 < pdf, portrait>
 < odf:odt, open document text>
 < xhtml scroll>
 < xml, sax>
 < xml, dom>
 < plain text utf-8>
 < concordance>
 < dcc, document content certificate (digests)>
 <markup source text>
 <markup source (zipped) pod>

660 **“Gnu Public License 2”, (GPL 2) Free Software
 Foundation**

661 <“Gnu Public License 2”, (GPL 2) Free Software Foundation>

⁴¹<http://www.sisudoc.org/sisu/alices_adventures_in_wonderland_and_through_the_looking_glass.lewis_carroll/sisu_manifest.html>

<document manifest> ⁴²
 < html, segmented text>
 < html, scroll, document in one>
 < epub>
 < pdf, landscape>
 < pdf, portrait>
 < odf:odt, open document text>
 < xhtml scroll>
 < xml, sax>
 < xml, dom>
 < plain text utf-8>
 < concordance>
 < dcc, document content certificate (digests)>
 <markup source text>
 <markup source (zipped) pod>

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

662

<“Gnu Public License 3 - Third discussion draft”, (GPL v3 draft3) Free Software
 Foundation>

663

<document manifest> ⁴³
 < html, segmented text>
 < html, scroll, document in one>
 < epub>
 < pdf, landscape>
 < pdf, portrait>
 < odf:odt, open document text>
 < xhtml scroll>
 < xml, sax>
 < xml, dom>

⁴²<http://www.sisudoc.org/sisu/gpl2.fsf/sisu_manifest.html>

⁴³<http://www.sisudoc.org/sisu/gpl3_draft3.fsf/sisu_manifest.html>

< plain text utf-8>
 < concordance>
 < dcc, document content certificate (digests)>
 <markup source text>
 <markup source (zipped) pod>

664 **“Debian Social Contract”**

665

<“Debian Social Contract”>
 <document manifest> ⁴⁴
 < html, segmented text>
 < html, scroll, document in one>
 < epub>
 < pdf, landscape>
 < pdf, portrait>
 < odf:odt, open document text>
 < xhtml scroll>
 < xml, sax>
 < xml, dom>
 < plain text utf-8>
 < concordance>
 < dcc, document content certificate (digests)>
 <markup source text>
 <markup source (zipped) pod>

666 **“Debian Constitution v1.3”, (simple/default markup)**

667

<“Debian Constitution v1.3”>

⁴⁴<http:
 //www.sisudoc.org/sisu/debian_social_contract_v1.1/sisu_manifest.html>

<document manifest> ⁴⁵
 < html, segmented text>
 < html, scroll, document in one>
 < epub>
 < pdf, landscape>
 < pdf, portrait>
 < odf:odt, open document text>
 < xhtml scroll>
 < xml, sax>
 < xml, dom>
 < plain text utf-8>
 < concordance>
 < dcc, document content certificate (digests)>
 <markup source text>
 <markup source (zipped) pod>

“Debian Constitution v1.3”, (markup adjusted for output to more closely match the original) 668

669

<“Debian Constitution v1.3”, (markup adjusted for output to more closely match the original)>

<document manifest> ⁴⁶
 < html, segmented text>
 < html, scroll, document in one>
 < epub>
 < pdf, landscape>
 < pdf, portrait>
 < odf:odt, open document text>
 < xhtml scroll>

⁴⁵<http:
 //www.sisudoc.org/sisu/debian_constitution_v1.3/sisu_manifest.html>

⁴⁶<http://www.sisudoc.org/sisu/debian_constitution_v1.3.adjusted/sisu_manifest.html>

```

< xml, sax>
< xml, dom>
< plain text utf-8>
< concordance>
< dcc, document content certificate (digests)>
<markup source text>
<markup source (zipped) pod>

```

670 **“Debian Constitution v1.2”, (simple/default markup)**

671

```

<"Debian Constitution v1.2 (more translations)">
  <document manifest> 47
  < html, segmented text>
  < html, scroll, document in one>
  < epub>
  < pdf, landscape>
  < pdf, portrait>
  < odf:odt, open document text>
  < xhtml scroll>
  < xml, sax>
  < xml, dom>
  < plain text utf-8>
  < concordance>
  < dcc, document content certificate (digests)>
  <markup source text>
  <markup source (zipped) pod>

```

672 **“Debian Constitution v1.2”, (markup adjusted for output to more closely match the original)**

673

```

<"Debian Constitution (more translations)", (markup adjusted for output to
more closely match the original)>
  <document manifest> 48
  < html, segmented text>
  < html, scroll, document in one>
  < epub>
  < pdf, landscape>
  < pdf, portrait>
  < odf:odt, open document text>
  < xhtml scroll>
  < xml, sax>
  < xml, dom>
  < plain text utf-8>
  < concordance>
  < dcc, document content certificate (digests)>
  <markup source text>
  <markup source (zipped) pod>

```

“A Uniform Sales Terminology”, Vikki Rogers and Albert Kritzer 674

675

```

<"A Uniform Sales Terminology", Vikki Rogers and Albert Kritzer>
  <document manifest> 49
  < html, segmented text>

```

⁴⁷http://www.sisudoc.org/sisu/debian_constitution_v1.2/sisu_manifest.html

⁴⁸http://www.sisudoc.org/sisu/debian_constitution_v1.2.adjusted/sisu_manifest.html

⁴⁹http://www.sisudoc.org/sisu/a_uniform_international_sales_terminology.vikki_rogers.and.albert_kritzer/sisu_manifest.html

```

< html, scroll, document in one>
< epub>
< pdf, landscape>
< pdf, portrait>
< odf:odt, open document text>
< xhtml scroll>
< xml, sax>
< xml, dom>
< plain text utf-8>
< concordance>
< dcc, document content certificate (digests)>
<markup source text>
<markup source (zipped) pod>

```

676 **“The Autonomous Contract” 1997 - markup sample**

677

```

<”The Autonomous Contract” 1997 - markup sample>
<document manifest>50
< html, segmented text>
< html, scroll, document in one>
< epub>
< pdf, landscape>
< pdf, portrait>
< odf:odt, open document text>
< xhtml scroll>
< xml, sax>
< xml, dom>
< plain text utf-8>
< concordance>
< dcc, document content certificate (digests)>
<markup source text>
<markup source (zipped) pod>

```

⁵⁰http://www.sisudoc.org/sisu/the_autonomous_contract.amissah.19970710/sisu_manifest.html

“The Autonomous Contract Revisited” - markup sample

678

```

<”The Autonomous Contract Revisited” - markup sample>51
<document manifest>52
< html, segmented text>
< html, scroll, document in one>
< epub>
< pdf, landscape>
< pdf, portrait>
< odf:odt, open document text>
< xhtml scroll>
< xml, sax>
< xml, dom>
< plain text utf-8>

```

679

⁵¹http://www.jus.uio.no/sisu/autonomy_markup0/toc.html [U+258C] alternative markup variations revolving around endnotes [U+258C] (i) as above, markup with embedded endnotes, and header list of words/phrases to emphasise [U+258C] http://www.jus.uio.no/sisu/sample/syntax/autonomy_markup0.sst.html [U+258C] http://www.jus.uio.no/sisu/sample/markup/autonomy_markup0.sst [U+258C] (ii) Again markup with embedded endnotes, but font faces changed within paragraphs rather than in header as in i [U+258C] http://www.jus.uio.no/sisu/sample/syntax/autonomy_markup1.sst.html [U+258C] http://www.jus.uio.no/sisu/sample/markup/autonomy_markup1.sst [U+258C] (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 [U+258C] http://www.jus.uio.no/sisu/sample/markup/autonomy_markup2.sst [U+258C] (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.

⁵²http://www.sisudoc.org/sisu/autonomy_markup0/sisu_manifest.html

< concordance>
 < dcc, document content certificate (digests)>
 <markup source text>
 <markup source (zipped) pod>

<markup source text>
 <markup source (zipped) pod>

680 *“United Nations Convention on Contracts for the
 International Sale of Goods”*

/PECL/ the “Principles of European Contract Law”

682

681

<“United Nations Convention on Contracts for the International Sale of
 Goods”>⁵³

<“Principles of European Contract Law”>

683

<document manifest>⁵⁴
 < html, segmented text>
 < html, scroll, document in one>
 < epub>
 < pdf, landscape>
 < pdf, portrait>
 < odf:odt, open document text>
 < xhtml scroll>
 < xml, sax>
 < xml, dom>
 < plain text utf-8>
 < concordance>
 < dcc, document content certificate (digests)>

<document manifest>⁵⁵
 < html, segmented text>
 < html, scroll, document in one>
 < epub>
 < pdf, landscape>
 < pdf, portrait>
 < odf:odt, open document text>
 < xhtml scroll>
 < xml, sax>
 < xml, dom>
 < plain text utf-8>
 < concordance>
 < dcc, document content certificate (digests)>
 <markup source text>
 <markup source (zipped) pod>

⁵³<http://www.jus.uio.no/sisu/un_contracts_international_sale_of_goods_convention_1980/toc.html> [U+258C] 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). [U+258C] 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).

⁵⁴<http://www.sisudoc.org/sisu/un_contracts_international_sale_of_goods_convention_1980/sisu_manifest.html>

21.3 SQL - PostgreSQL, SQLite

684

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

685

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

686

<English documents matching Linux OR Debian>

687

<GPL OR Stallman>

688

<invention OR innovation>

689

<copyright in English language documents>

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

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

693 <Linux OR Debian>

694 <GPL OR Stallman>

695 <invention OR innovation in English language>

696 <copyright in English language documents>

697 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.⁵⁶

698 21.4 Lex Mercatoria as an example

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

700 <<http://www.lexmercatoria.org/>>⁵⁷

701 or perhaps:

702 <<http://lexmercatoria.org/treaties.and.organisations/lm.chronological>>⁵⁸

⁵⁵ <http://www.sisudoc.org/sisu/eu_contract_principles_parts_1_to_3_2002/sisu_manifest.html>

⁵⁶ 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.

⁵⁷ <<http://www.jus.uio.no/lm/index>>

⁵⁸ <<http://www.jus.uio.no/lm/treaties.and.organisations/lm.chronological>>

690 **21.5 For good measure the markup for a document with lots of (simple) tables** 703

SiSU is not optimised for table making, but does handle simple tables. 704

Γ <[Output of table file example](#)>⁵⁹ 705

21.6 And a link to the output of a reported case 706

<<http://www.jus.uio.no/lm/england.fothergill.v.monarch.airlines.hl.1980/toc.html>> 707

708 **22. A Checklist of Output Features**709 This table gives an indication of the features that are available for
various forms of output of SiSU .710 **sisu-2.0.0 on 2010-03-06**

711

feature	txt	ltx/pdf	HTML	EPUB	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
image caption	-	*	*	*					
table of contents		*	*	*	*	*	.		
page header/footer?	-	*	*	*	*	*	t		
line break	*	*	*	*	*	*	*		
page break		*					*		
segments			*	*					
skins	*	*	*	*	*	*			
ocn	.	*	*	*	*	*	-?	*	*
auto-heading numbers	*	*	*	*	*	*	*	*	*

feature	txt	ltx/pdf	HTML	EPUB	XML/s	XML/d	ODF	SQLite	pgSQL
minor list numbering	*	*	*	*	*	*	*	*	*
special characters

712

sisu-1.0.0 on 2009-10-28

713

feature	txt	ltx/pdf	HTML	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
image caption	-	*	*
table of contents	.	*	*	*	*	.	.	.
page header/footer?	-	*	*	*	*	t	.	.
line break	*	*	*	*	*	*	.	.
page break	.	*	.	.	.	*	.	.
segments	.	.	*
skins	*	*	*	*	*	.	.	.
ocn	.	*	*	*	*	-?	*	*

feature	txt	ltx/pdf	HTML	XML/s	XML/d	ODF	SQLite	pgSQL
auto-heading numbers	*	*	*	*	*	*	*	*
minor list numbering	*	*	*	*	*	*	*	*
special characters

714

sisu-0.36.6 on 2006-01-23

715

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	.	*	*	*	*	*	-?	*	*

feature	txt	ltx/pdf	HTML	XHTML	XML/s	XML/d	ODF	SQLite	pgsql	717
auto-heading numbers	*	*	*	*	*	*	*	*	*	717
minor list numbering	*	*	*	*	*	*	*	*	*	718
special characters	719
Done										
* yes/done										
. partial										
- not available/appropriate										
Not Done										
T task todo										
t lesser task/todo										
not done										

2.2 SiSU Search - Introduction 717

SiSU output can easily and conveniently be indexed by a number of standalone indexing tools, such as Lucene, Hyperestraier. 718

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. 719

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. 720

716

721 24. SQL

722 24.1 populating SQL type databases

723 **SiSU** feeds `sisu` markupd documents into sql type databases PostgreSQL⁶⁰ and/or SQLite⁶¹ database together with information related to document structure.

724 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:

725 Γ one containing semantic (and other) headers, including, title, author, subject, (the Dublin Core...);

726 Γ 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

727 Γ a third containing endnotes cross-referenced back to the paragraph from which they are referenced (both in formatted and clean text versions for searching).

728 Γ 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.

729 There is of course the possibility to add further structures.

730 At this level **SiSU** loads a relational database with documents

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>

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).

731

732 **25. Postgresql**733 **25.1 Name**

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

735 **25.2 Description**

736 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).

737 **25.3 Synopsis**

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

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

740 **25.4 Commands**

741 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

742 **-D** or **-pgsql** may be used interchangeably.

743 **25.4.1 create and destroy database**

744 **-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

745

sisu -D -create

creates database tables where no database tables existed before

746

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).

747

sisu -D -recreate

destroys existing database and builds a new empty database structure

748

25.4.2 import and remove documents

749

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).

750

sisu -D -update -v [filename/wildcard]

updates file contents in database

751

sisu -D -remove -v [filename/wildcard]

removes specified document from postgresql database.

752

753 **26. Sqlite**754 **26.1 Name**

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

756 **26.2 Description**

757 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).

758 **26.3 Synopsis**

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

761 **26.4 Commands**

762 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

763 **-d** or **-sqlite** may be used interchangeably.

764 **26.4.1 create and destroy database**

765 **-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 be-
fore

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 struc-
ture

770 **26.4.2 import and remove documents**

sisu -d -import -v [filename/wildcard]

populates database with the contents of the file. Imports docu-
ments(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.

774 **27. Introduction**775 **27.1 Search - database frontend sample, utilising database and SiSU features, including object citation numbering (backend currently PostgreSQL)**

776 <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

777 **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, EPUB, 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.

778 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

779 **sisu -F --webserv-webrick**

builds a cgi web search frontend for the database created

780 The following is feedback on the setup on a machine provided by

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.

⁶⁴of 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.

the help command:

```

sisu -help sql 781
Postgresql 782
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 783

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] 784

785 **27.2 Search Form**

sisu -F 786

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

sisu -F --webserv-webrick 787

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

sisu -Fv 788

as above, and provides some information on setting up hyperstraier

sisu -W		
starts the webrick server which should be available wherever sisu is properly installed		
790 The generated search form must be copied manually to the web-server directory as instructed		
	789	28. Hyperestraier
		791
		See the documentation for hyperestraier:
		792
		<code><http://hyperestraier.sourceforge.net/></code>
		793
		<code>/usr/share/doc/hyperestraier/index.html</code>
		794
		<code>man estcmd</code>
		795
		on <code>sisu_hyperestraier</code> :
		796
		<code>man sisu_hyperestraier</code>
		797
		<code>/usr/share/doc/sisu/sisu-markup/sisu_hyperestraier/index.-html</code>
		798
		NOTE: the examples that follow assume that sisu output is placed in the directory <code>/home/ralph/sisu_www</code>
		799
		(A) to generate the index within the webserver directory to be indexed:
		800
		<code>estcmd gather -sd [index name] [directory path to index]</code>
		801
		the following are examples that will need to be tailored according to your needs:
		802
		<code>cd /home/ralph/sisu_www</code>
		803
		<code>estcmd gather -sd casket /home/ralph/sisu_www</code>
		804
		you may use the ‘find’ command together with ‘egrep’ to limit indexing to particular document collection directories within the web server directory:
		805
		<code>find /home/ralph/sisu_www -type f egrep ‘/home/ralph/-sisu_www/sisu/.+?.html\$’ estcmd gather -sd casket -</code>
		806
		Check which directories in the webserver/output directory (<code>~/sisu_www</code> or elsewhere depending on configuration) you wish to include in the search index.
		807
		As sisu duplicates output in multiple file formats, it it is probably
		808

preferable to limit the estrailer 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.

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

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

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

812 (B) to set up the search form

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

```
814 sudo cp -vi /usr/lib/estraier/estseek.cgi /usr/lib/cgi-bin
815 sudo chmod -v 755 /usr/lib/cgi-bin/estseek.cgi
816 sudo cp -v /usr/share/hyperestraier/estseek.* /usr/lib/cgi-bin
817 [see estrailer documentation for paths]
```

818 (ii) edit estseek.conf, with attention to the lines starting ‘index-name:’ and ‘replace:’:

```
819 indexname: /home/ralph/sisu_www/casket
820 replace: ^ file:///home/ralph/sisu_www<{!}>
821 replace: /index.html?${!}/
```

822 (C) to test using webrick, start webrick:

```
823 sisu -W
```

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

29. sisu_webrick 825

29.1 Name 826

SiSU - Structured information, Serialized Units - a document publishing system 827

29.2 Synopsis 828

sisu_webrick [port] 829

or 830

sisu -W [port] 831

29.3 Description 832

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). 833

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). 834

29.4 Summary of man page 835

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] 836

where no port is given and settings are unchanged the default port is 8081 837

838 **29.5 Document processing command flags**

839 `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

840 **29.6 Further information**

841 For more information on **SiSU** see: [<http://www.sisudoc.org/>](http://www.sisudoc.org/) or
[<http://www.jus.uio.no/sisu>](http://www.jus.uio.no/sisu)

842 or `man sisu`

843 **29.7 Author**

844 Ralph Amissah [<ralph@amissah.com>](mailto:ralph@amissah.com) or [<ralph.amissah@gmail.com>](mailto:ralph.amissah@gmail.com)

845 **29.8 SEE ALSO**

846 `sisu(1)`

847 `sisu_vim(7)`

30. Remote Source Documents 848

SiSU processing instructions can be run against remote source 849
 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 850

SiSU can be run against source text files on a remote machine, 851
 provide the processing instruction and the url. The source file and
 any associated parts (such as images) will be downloaded and gener-
 ated locally.

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

 852

Any of the source documents in the `sisu` examples page can be used 853
 in this way, see [<http://www.jus.uio.no/sisu/SiSU/examples.html>](http://www.jus.uio.no/sisu/SiSU/examples.html) and use
 the url to the `.sst` for the desired document.

NOTE: to set up a remote machine to serve **SiSU** documents in 854
 this way, images should be in the directory relative to the document
 source `../_sisu/image`

sisupod - zipped sisu files 855

A `sisupod` is the zipped content of a `sisu` marked up text or texts and 856
 any other associated parts to the document such as images.

SiSU can be run against a `sisupod` on a (local or) remote machine, 857
 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]
```

 858

Any of the source documents in the `sisu` examples page can be used 859
 in this way, see [<http://www.jus.uio.no/sisu/SiSU/examples.html>](http://www.jus.uio.no/sisu/SiSU/examples.html) and use
 the url for the desired document.

860 Remote Document Output

31. Remote Output 861

Once properly configured **SiSU** output can be automatically posted once generated to a designated remote machine using either rsync, or scp. 862

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. 863

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

 864

31.1 commands 865

-R [filename/wildcard] 866

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] 867

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

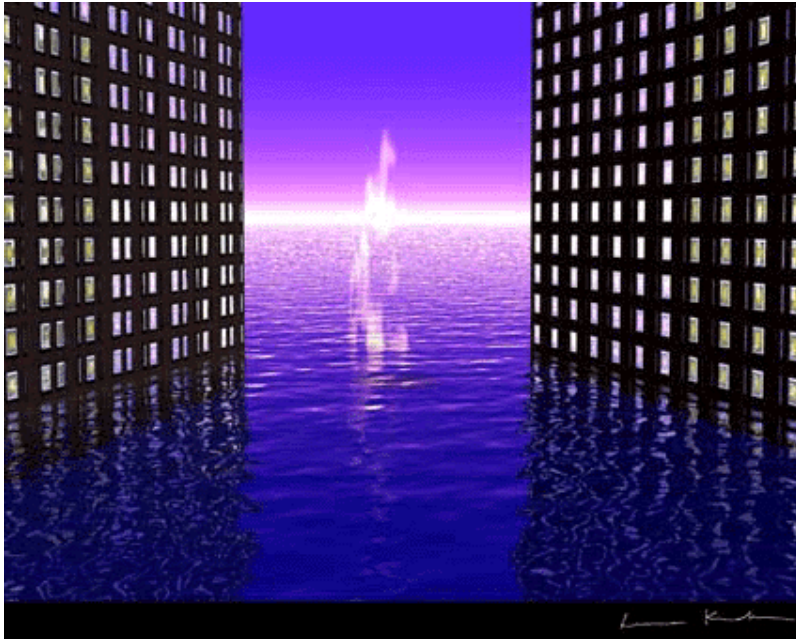
31.2 configuration 868

[expand on the setting up of an ssh-agent / keychain] 869

870 **32. Remote Servers**

871 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.

872	Download information		
		33. Download SiSU - Linux/Unix	873
		SiSU Current Version - Linux/Unix	874
		Source (tarball tar.gz)	875
		Download the latest version of SiSU (and SiSU markup samples):	876
		Γ <code><sisu_3.0.4.orig.tar.gz (2011-03-11:10/5)></code> ⁶⁵	877
		Γ <code><sisu-markup-samples_3.0.0.orig.tar.gz (of 2011-02-16:07/3)></code> ⁶⁶	878
		For installation notes see <code><http://sisudoc.org/sisu/sisu_manual/installation.html></code>	879
		For more general use see <code><http://sisudoc.org/sisu/sisu_manual></code>	880
		For changelogs see <code><http://www.jus.uio.no/sisu/SiSU/changelog.html></code>	881
		⁶⁵ <code><http://www.jus.uio.no/sisu/pkg/src/sisu_3.0.4.orig.tar.gz> 145c409526b26cb0a14b43f4c46219fb828dc41c8211d8f77bad486a98300678 1920526</code>	
		⁶⁶ <code><http://www.jus.uio.no/sisu/pkg/src/sisu-markup-samples_3.0.0.orig.tar.gz> 999f3cc572d0558a6af4539db0c51691dcff3371d4f92e096cbf5835806aeced4 8446814</code>	



882

67

883 Git (source control management)

884 Git repository currently at:

885 `Γ git clone git://git.sisudoc.org/git/code/sisu.git`

886 `Γ <http://git.sisudoc.org/?p=code/sisu.git;a=summary>`

887 On using git, see

⁶⁷Reproduced 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 <<http://www.jus.uio.no/sisu/SiSU/examples.html#sample>>

Git documentation: <Git User's Manual>, ⁶⁸ <Everyday GIT 888
With 20 Commands Or So>, ⁶⁹ <A tutorial introduction to git>, ⁷⁰
<A tutorial introduction to git: part two> ⁷¹

User contributed texts: <The Git Community Book>, ⁷² <Git 889
Magic>, ⁷³ <Git From the Bottom Up (pdf)> ⁷⁴

Debian 890

This section contains information on the latest **SiSU** release. For 891
installation notes see <http://sisudoc.org/sisu/sisu_manual/installation.html>

SiSU is updated fairly regularly in **Debian** testing and unstable, 892
and should be available therefrom.

To add this archive, should you still choose to do so, add the fol- 893
lowing lines to your `/etc/apt/sources.list`

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

[U+3014]b[Source]b[U+3015]

<sisu.3.0.4.orig.tar.gz> ⁷⁵

<sisu.3.0.4-1.debian.tar.gz> ⁷⁶

⁶⁸<<http://www.kernel.org/pub/software/scm/git/docs/user-manual.html>>

⁶⁹<<http://www.kernel.org/pub/software/scm/git/docs/everyday.html>>

⁷⁰<<http://www.kernel.org/pub/software/scm/git/docs/v1.4.4.4/tutorial.html>>

⁷¹<<http://www.kernel.org/pub/software/scm/git/docs/v1.4.4.4/tutorial-2.html>>

⁷²<<http://book.git-scm.com/index.html>>

⁷³<<http://www.cs-students.stanford.edu/~blynn/gitmagic/>>

⁷⁴<http://www.newartisans.com/blog_assets/git.from.bottom.up.pdf>

⁷⁵<http://www.jus.uio.no/sisu/pkg/src/sisu_3.0.4.orig.tar.gz>

⁷⁶<http://www.jus.uio.no/sisu/pkg/src/sisu_3.0.4-1.diff.gz>

<sisu_3.0.4-1.dsc> ⁷⁷

[U+3014]bDebsb [U+3015]

<sisu_3.0.4-1_all.deb> ⁷⁸

<sisu-complete_3.0.4-1_all.deb> ⁷⁹

<sisu-pdf_3.0.4-1_all.deb> ⁸⁰

<sisu-postgresql_3.0.4-1_all.deb> ⁸¹

<sisu-sqlite_3.0.4-1_all.deb> ⁸²

⁷⁷<http://www.jus.uio.no/sisu/pkg/src/sisu_3.0.4-1.dsc>

[U+258C]4886196e536e3ad48b29d27b1f0664f77b58ee4b4b21e41c0351c6b3a33dd1561216

⁷⁸<http://www.jus.uio.no/sisu/archive/pool/main/s/sisu/sisu_3.0.4-1_all.deb>

[U+258C]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 [U+258C]Depends: ruby (≥1.8.2), libwebrick-ruby [U+258C]Recommends: sisu-pdf, sisu-sqlite, sisu-postgresql, sisu-examples, vim-sisu, libmagick-ruby, trang, tidy, libtidy, libexml-ruby, zip, unzip, openssl

⁷⁹<http://www.jus.uio.no/sisu/archive/pool/main/s/sisu/sisu-complete_3.0.4-1_all.deb>

[U+258C]a package that pulls in other packages to build the whole of sisu (excluding sisu-examples) [U+258C]Depends: ruby (≥1.8.2), sisu, sisu-pdf, sisu-postgresql, sisu-remote, sisu-sqlite, vim-sisu [U+258C]Recommends: sisu-examples

⁸⁰<http://www.jus.uio.no/sisu/archive/pool/main/s/sisu/sisu-pdf_3.0.4-1_all.deb>

[U+258C]dependencies used by sisu to produce pdf from LaTeX generated [U+258C]Depends: sisu, tetex-bin, tetex-extra, latex-ucs [U+258C]Suggests: evince, xpdf

⁸¹<http://www.jus.uio.no/sisu/archive/pool/main/s/sisu/sisu-postgresql_3.0.4-1_all.deb>

[U+258C]dependencies used by sisu to populate postgresql database (further configuration is necessary) [U+258C]Depends: sisu, postgresql-8.1, libdbi-ruby, libdbm-ruby, libdbd-pg-ruby [U+258C]Suggests: pgaccess, libdbd-pgsql, postgresql-contrib-8.1

⁸²<http://www.jus.uio.no/sisu/archive/pool/main/s/sisu/sisu-sqlite_3.0.4-1_all.deb>

[U+258C]dependencies used by sisu to populate sqlite database [U+258C]Depends: sisu, sqlite, libdbi-ruby, libdbm-ruby, libdbd-sqlite-ruby [U+258C]Suggests: libdbd-sqlite

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

Γ <sisu-markup-samples_3.0.0-1_all.deb> ⁸³

Γ <sisu-markup-samples_3.0.0-1.dsc> ⁸⁴

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 from the source file using Alien.⁸⁵ Dependencies are not handled, not even that of the essential **Ruby**. In the Howto section see the note on installing **SiSU** on Fedora 12 (2010-01-12).

sudo rpm -i [package name]

⁸³<http://www.jus.uio.no/sisu/archive/pool/non-free/s/sisu-markup-samples/sisu-markup-samples_3.0.0-1_all.deb>

marked up documents and other examples related to sisu, a larger package containing a number of texts

Depends: sisu

⁸⁴<http://www.jus.uio.no/sisu/pkg/src/sisu-markup-samples_3.0.0-1.dsc> 7d9d434c74a1e96da3732e420d483466d7ca1266d4e6fba4bf721b9e3f73aad 1307 sisu-markup-samples_3.0.0-1.dsc

⁸⁵<<http://www.kitenet.net/programs/alien/>>

910 Γ <sisu-1.0.3-2.noarch.rpm> ⁸⁶

911 Γ <sisu-markup-samples_2.0.3.orig-2.noarch.rpm> ⁸⁷

912 For changelogs see:

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

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

⁸⁶ <<http://www.jus.uio.no/sisu/pkg/rpm/sisu-1.0.3-2.noarch.rpm>>

*

created using alien

⁸⁷ <http://www.jus.uio.no/sisu/pkg/rpm/sisu-markup-samples_2.0.3.orig-2.noarch.rpm>

<http://www.jus.uio.no/sisu/archive/pool/non-free/s/sisu-markup-samples/sisu-markup-samples_2.0.3-1_all.deb>

created using: alien -r sisu_1.0.3-1_all.deb

915	Installation	
	34. Installation	916
	See the < download pages > ⁸⁸ for information related to installation.	917
	34.1 Debian	918
	SiSU is developed on Debian , and packages are available for Debian that take care of the dependencies encountered on installation.	919
	The package is divided into the following components:	920
	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	921
	sisu-complete, a dummy package that installs the whole of greater sisu as described below, apart from sisu-examples	922
	sisu-pdf, dependencies used by sisu to produce pdf from LaTeX generated	923
	sisu-postgresql, dependencies used by sisu to populate postgresql database (further configuration is necessary)	924
	sisu-remote, dependencies used to place sisu output on a remote server (further configuration is necessary)	925
	sisu-sqlite, dependencies used by sisu to populate sqlite database	926
	sisu-markup-samples, sisu markup samples and other miscellany (under Debian Free Software Guidelines non-free)	927
	SiSU is available off < Debian Unstable and Testing > ⁸⁹ install it	928
	⁸⁸ < http://www.jus.uio.no/sisu/SiSU/download.html >	
	⁸⁹ < http://packages.qa.debian.org/s/sisu.html >	

using apt-get, aptitude or alternative **Debian** install tools. **SiSU** is currently comprised of eight packages.

929 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:

```
930 #/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
```

931 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.

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

```
933 #Using aptitude:

    aptitude update

    aptitude install sisu-complete sisu-markup-samples
934 Using apt-get

    apt-get update

    apt get install sisu-complete sisu-examples
```

935 34.2 Other Unix / Linux

936 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** .⁹⁰ 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:⁹¹

```
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,⁹² 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>⁹³ 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>⁹⁴

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:

⁹⁰Notes on dependencies are provided in the section that follows

⁹¹This 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.

⁹²a Rantfile has been configured to do post installation setup

⁹³<<http://make.rubyforge.org/>> <<http://make.rubyforge.org/>>
<http://rubyforge.org/frs/?group_id=615>

⁹⁴<<http://make.rubyforge.org/>> <<http://make.rubyforge.org/>>
<http://rubyforge.org/frs/?group_id=615>

945 ruby install -T
 946 To install as root type:
 947 ruby install
 948 For a minimal install type:
 949 ruby install base
 950 **installation with setup.rb**
 951 `<setup.rb>`⁹⁵ is provided the package and will install **SiSU**⁹⁶ installation is a 3 step process⁹⁷ the following string assumes you are in the package directory and that you have root as sudo:
 952 ruby setup.rb config && ruby setup.rb setup && sudo ruby setup.rb install
 953 **installation of rpm**
 954 The RPM is generated from the source file using Alien.⁹⁸ Dependencies are not handled, not even that of the essential **Ruby** .

⁹⁵<http://i.loveruby.net/en/projects/setup/>

⁹⁶Minero Aoki

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

⁹⁷Installation instructions

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

⁹⁸<http://www.kitenet.net/programs/alien/>

35. SiSU Components, Dependencies and Notes 955

The dependency lists are from the **Debian** control file for **SiSU** version 0.36, and may assist in building **SiSU** on other distributions. 956

35.1 sisu 957

Γ 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 958

Depends: on ruby (≥1.8.2), libwebrick-ruby 959

Recommends: sisu-pdf, sisu-sqlite, sisu-postgresql, sisu-examples, librmagick-ruby, trang, tidy, libtidy, librexml-ruby, zip, unzip, openssl 960

initialise directory 961

sisu -CC 962

html 963

sisu -hv [filename/wildcard] 964

sisu -Hv [filename/wildcard] 965

LaTeX (but sisu-pdf dependencies required to convert that to pdf) 966

sisu -pv [filename/wildcard] 967

plain text Unix with footnotes 968

sisu -av [filename/wildcard] 969

plain text Dos with footnotes 970

sisu -Av [filename/wildcard] 971

972	plain text Unix with endnotes	996	sisu -Uv [filename/wildcard]	
973	sisu -ev [filename/wildcard]		toggle screen colour	997
974	plain text Dos with endnotes		sisu -cv[and processing flags] [filename/wildcard]	998
975	sisu -Ev [filename/wildcard]		verbose mode	999
976	openoffice odt		sisu -v[and processing flags] [filename/wildcard]	1000
977	sisu -ov [filename/wildcard]		sisu -V[and processing flags] [filename/wildcard]	1001
978	xhtml		quiet mode	1002
979	sisu -bv [filename/wildcard]		sisu -q[and processing flags] [filename/wildcard]	1003
980	XML SAX		maintenance mode, intermediate files kept -M	1004
981	sisu -xv [filename/wildcard]		sisu -Mv[and other flags] [filename/wildcard]	1005
982	XML DOM		[the -v is for verbose]	1006
983	sisu -Xv [filename/wildcard]		start the webrick server	1007
984	wordmap (a rudimentary index of content)		sisu -W	1008
985	sisu -wv [filename/wildcard]			
986	document content certificate			
987	sisu -Nv [filename/wildcard]		35.2 sisu-complete	1009
988	placement of sourcefile in output directory		Γ a dummy package that installs the whole SiSU , apart from	1010
989	sisu -sv [filename/wildcard]		sisu-examples	
990	creation of source tarball with images, and placement of source tarball in ouput directory		*Depends:* ruby (<i>i</i> =1.8.2), sisu, sisu-pdf, sisu-postgresql, sisu-remote, sisu-sqlite	1011
991	sisu -Sv [filename/wildcard]		*Recommends:* sisu-examples	1012
992	manifest of output produced (polls output directory and provides links to existing output)			
993	sisu -yv [filename/wildcard]		35.3 sisu-examples	1013
994	url for output files -u -U		Γ installs sisu markup samples and other miscelleny	1014
995	sisu -uv[and other flags] [filename/wildcard]		*Depends:* sisu	1015

	35.4 sisu-pdf	1037	sisu -Dv update [filename/wildcard]	
1017	Γ dependencies used by sisu to produce pdf from LaTeX generated		[the -v is for verbose]	1038
1018	*Depends:* sisu, tetex-bin, tetex-extra, latex-ucs		The following are available without installation of the sisu- postgresql component, but are of interest in this context	1039
1019	*Suggests:* evince, xpdf		generate a sample database query form for use with webserver on port 80	1040
1020	converts sisu LaTeX produced to pdf		sisu -F	1041
1021	sisu -pv [filename/wildcard]		or for use with webrick server	1042
1022	[the -v is for verbose]		sisu -F webrick	1043
1023	35.5 sisu-postgresql		to start webrick server	1044
1024	Γ dependencies used by sisu to populate postgresql database (fur- ther configuration is necessary)		sisu -W	1045
1025	*Depends:* sisu, postgresql-8.1, libdbi-ruby, libdbm-ruby, libdbd- pg-ruby		35.6 sisu-remote	1046
1026	*Suggests:* pgaccess, libdbd-pgsql, postgresql-contrib-8.1		Γ dependencies used to place sisu output on a remote server (fur- ther configuration is necessary)	1047
1027	installs dependencies for sisu to work with and populate postgresql database		scp	1048
1028	create database		sisu -vr[and processing flags] [filename/wildcard]	1049
1029	sisu -Dv createall		rsync	1050
1030	drop database		sisu -vR[and processing flags] [filename/wildcard]	1051
1031	sisu -Dv dropall		[the -v is for verbose]	1052
1032	import content		*Depends:* sisu, rsync, openssh-client lsh-client, keychain	1053
1033	sisu -Div [filename/wildcard]		35.7 sisu-sqlite	1054
1034	sisu -Dv import [filename/wildcard]		Γ dependencies used by sisu to populate sqlite database	1055
1035	update content			
1036	sisu -Duv [filename/wildcard]			

<p>*Depends:* sisu, sqlite, libdbi-ruby, libdbm-ruby, libdbd-sqlite-ruby</p> <p>*Suggests:* libdbd-sqlite</p> <p>1058 installs dependencies for sisu to work with and populate sqlite database</p> <p>1059 create database</p> <p>1060 sisu -dv createall</p> <p>1061 drop database</p> <p>1062 sisu -dv dropall</p> <p>1063 update content</p> <p>1064 sisu -div [filename/wildcard]</p> <p>1065 sisu -dv import [filename/wildcard]</p> <p>1066 update content</p> <p>1067 sisu -duv [filename/wildcard]</p> <p>1068 sisu -dv update [filename/wildcard]</p> <p>1069 [the -v is for verbose]</p> <p>1070 The following are available without installation of the sisu-sqlite component, but are of interest in this context</p> <p>1071 generate a sample database query form for use with webserver on port 80</p> <p>1072 sisu -F</p> <p>1073 or for use with webrick server</p> <p>1074 sisu -F webrick</p> <p>1075 to start webrick server</p> <p>1076 sisu -W</p>	<p>1056 36. Quickstart - Getting Started Howto 1077</p> <p>1057 36.1 Installation 1078</p> <p>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. 1079</p> <p>36.1.1 Debian Installation 1080</p> <p>SiSU is available directly from the Debian Sid and testing archives (and possibly Ubuntu), assuming your <code>/etc/apt/sources.list</code> is set accordingly: 1081</p> <pre>aptitude update 1082 aptitude install sisu-complete</pre> <p>The following <code>/etc/apt/sources.list</code> setting permits the download of additional markup samples: 1083</p> <pre>#/etc/apt/sources.list 1084 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</pre> <p>The aptitude commands become: 1085</p> <pre>aptitude update 1086 aptitude install sisu-complete sisu-markup-samples</pre> <p>If there are newer versions of SiSU upstream of the Debian archives, they will be available by adding the following to your <code>/etc/apt/sources.list</code> 1087</p> <pre>#/etc/apt/sources.list 1088 deb http://www.jus.uio.no/sisu/archive unstable main non-free</pre>
--	--

```

1089     deb-src http://www.jus.uio.no/sisu/archive unstable main
1090     non-free
1091 repeat the aptitude commands
1092     aptitude update
1093     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.

1092 36.1.2 RPM Installation

1093 RPMs are provided though untested, they are prepared by running
alien against the source package, and against the debs.

1094 They may be downloaded from:

```
1095 <http://www.jus.uio.no/sisu/SiSU/download.html#rpm>
```

1096 as root type:

```
1097 rpm -i [rpm package name]
```

1098 36.1.3 Installation from source

1099 To install **SiSU** from source check information at:

```
1100 <http://www.jus.uio.no/sisu/SiSU/download.html#current>
```

1101 Γ download the source package

1102 Γ Unpack the source

1103 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:

1104 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, EPUB, 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)⁹⁹ 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:

```

1107     ruby setup.rb config
1108     ruby setup.rb setup
1109     #[and as root:]
1110     ruby setup.rb install

```

further information on setup.rb is available from:

```
1111 <http://i.loveruby.net/en/projects/setup/>
```

```
1112 <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:

```
1113     ruby install base
```

or for a more complete installation:

```
1114     ruby install
```

or

```
1115     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

⁹⁹There is nothing to stop MySQL support being added in future.

	external package dependencies, such as tetex-extra are not taken care of for you.	text format, plaintext, XHTML and various forms of XML, and OpenDocument text	
1119	Further information on “rant” is available from:		
1120	<code><http://make.rubyforge.org/></code>	36.2.2 LaTeX / pdf	1136
1121	<code><http://rubyforge.org/frs/?group_id=615></code>	Assuming a LaTeX engine such as tetex or texlive is installed with the required modules (done automatically on selection of sisu-pdf in Debian)	1137
1122	For a list of alternative actions you may type:	Having moved to the directory that contains the markup samples (see instructions above if necessary), choose a file and run sisu against it	1138
1123	<code>ruby install help</code>		
1124	<code>ruby install -T</code>		
1125	36.2 Testing SiSU, generating output		
1126	To check which version of sisu is installed:	<code>sisu -pv free_as_in_freedom.rms_and_free_software.sam_williams.sst</code>	1139
1127	<code>sisu -v</code>	<code>sisu -3 free_as_in_freedom.rms_and_free_software.sam_williams.sst</code>	1140
1128	Depending on your mode of installation one or a number of markup sample files may be found either in the directory:	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	1141
	...		
1129	OR		
	...		
1130	change directory to the appropriate one:	36.2.3 relational database - postgresql, sqlite	1142
1131	<code>cd /usr/share/doc/sisu/markup-samples/samples</code>	Relational databases need some setting up - you must have permission to create the database and write to it when you run sisu.	1143
1132	36.2.1 basic text, plaintext, html, XML, ODF, EPUB	Assuming you have the database installed and the requisite permissions	1144
1133	Having moved to the directory that contains the markup samples (see instructions above if necessary), choose a file and run sisu against it	<code>sisu -sqlite -recreate</code>	1145
		<code>sisu -sqlite -v -import free_as_in_freedom.rms_and_free_software.sam_williams.sst</code>	
1134	<code>sisu -NhwoabxXyv free_as_in_freedom.rms_and_free_software.sam_williams.sst</code>	<code>sisu -pgsql -recreate</code>	1147
1135	this will generate html including a concordance file, opendocument	<code>sisu -pgsql -v -import free_as_in_freedom.rms_and_free_software.sam_williams.sst</code>	

36.3 Getting Help

36.3.1 The man pages

1151 Type:

1152 man sisu

1153 The man pages are also available online, though not always kept as
up to date as within the package itself:

1154 Γ <sisu.1> ¹⁰⁰

1155 Γ <sisu.8> ¹⁰¹

1156 Γ <man directory> ¹⁰²

1157 36.3.2 Built in help

1158 sisu -help

1159 sisu -help -env

1160 sisu -help -commands

1161 sisu -help -markup

1162 36.3.3 The home page

1163 <<http://www.sisudoc.org/>>

1164 <<http://www.jus.uio.no/sisu>>

1165 <<http://www.jus.uio.no/sisu/SiSU>>

1166 36.4 Markup Samples

1167 A number of markup samples (along with output) are available

¹⁰⁰<<http://www.jus.uio.no/sisu/man/sisu.1.html>>

¹⁰¹<<http://www.jus.uio.no/sisu/man/sisu.8.html>>

¹⁰²<<http://www.jus.uio.no/sisu/man>>

1149 off:

1150 <<http://www.jus.uio.no/sisu/SiSU/examples.html>>

Additional markup samples are packaged separately in the file: 1168 1169

On **Debian** they are available in non-free¹⁰³ to include them it is necessary to include non-free in your /etc/apt/source.list or obtain them from the sisu home site. 1170

¹⁰³the **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.

37. Getting Help 1172

An online manual of sorts should be available at: 1173

<http://www.jus.uio.no/sisu_manual/> 1174

The manual pages provided with **SiSU** are also available online, 1175
and there is an interactive help, which is being superseded by the
man page, and possibly some document which contains this com-
ponent.

37.1 SiSU “man” pages 1176

If **SiSU** is installed on your system usual man commands should 1177
be available, try:

man sisu 1178

The **SiSU** man pages can be viewed online at:¹⁰⁴ 1179

An online version of the sisu man page is available here: 1180

Γ <[various sisu man pages](#)> ¹⁰⁵ 1181

Γ <[sisu.1](#)> ¹⁰⁶ 1182

Γ <[sisu.8](#)> ¹⁰⁷ 1183

Γ <[sisu_webrick.1](#)> ¹⁰⁸ 1184

	37.2 SiSU built-in help		
1186	sisu -help		
1187	sisu -help [subject]		
1188	sisu -help env [for feedback on the way your system is setup with regard to sisu]		
1189	sisu -V [same as above command]		
1190	sisu -help commands		
1191	sisu -help markup		
1192	37.3 Command Line with Flags - Batch Processing		
1193	Running sisu (alone without any flags, filenames or wildcards) brings up the interactive help, as does any sisu command that is not recognised.		
1194	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.		
	¹⁰⁴ 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 > ¹⁰⁵ < http://www.jus.uio.no/sisu/man/ > ¹⁰⁶ < http://www.jus.uio.no/sisu/man/sisu.1.html > ¹⁰⁷ < http://www.jus.uio.no/sisu/man/sisu.8.html > ¹⁰⁸ < http://www.jus.uio.no/sisu/man/sisu_webrick.1.html >		
1185	38. Setup, initialisation		1195
	38.1 initialise output directory		1196
	Images, css files for a document directory are copied to their re- spective locations in the output directory.		1197
	while within your document markup/preparation directory, issue the following command		1198
	sisu -CC		1199
	38.1.1 Use of search functionality, an example using sqlite		1200
	SiSU can populate PostgreSQL and Sqlite databases and provides a sample search form for querying these databases.		1201
	This note provides an example to get you started and will use sqlite		1202
	It is necessary to:		1203
	(1) make sure the required dependencies have been installed		1204
	(2) have a directory with sisu markup samples that is writable		1205
	(3) use sisu to create a database		1206
	(4) use sisu tp populate a database		1207
	(5) use sisu to start the webrick (httpd) server		1208
	(6) use sisu to create a search form		1209
	(7) copy the search form to the cgi directory		1210
	(8) open up the form in your browser		1211
	(9) query the database using the search form		1212
	(1) make sure the required dependencies have been in- stalled		1213

1214	if you use Debian , the following command will install the required dependencies	1232	within the sisu-examples directory	
1215	aptitude install sisu-sqlite		sisu -F webserv=webrick	1233
1216	(2) have a directory with sisu markup samples that is writable		and follow the instructions provided	1234
1217	ideally copy the sisu-examples directory to your home directory (because the directory in which you run this example should be writable)		#here i run into a problem, you are working from a read only #directory..., not my usual mode of operation, to complete the example	1235
1218	cp -rv /usr/share/doc/sisu/markup-samples/samples .		#the following is necessary sudo touch sisu_sqlite.cgi sisu_pgsq.cgi	
1219	you are better off installing the package sisu-markup-samples which will make the following available		sudo -P chown \$USER sisu_sqlite.cgi sisu_pgsq.cgi	
1220	cp -rv /usr/share/doc/sisu/markup-samples-non-free/samples .		#now this should be possible: sisu -F webrick	1236
1221	(3) use sisu to create an sqlite database		(7) copy the search form to the cgi directory	1237
1222	within the sisu-examples directory		sisu -F webserv=webrick	1238
1223	sisu -dv createall		and follow the instructions provided	1239
1224	(4) use sisu tp populate a database with some text		(8) open up the form in your browser and query it	1240
1225	within the sisu-examples directory		url:	1241
1226	sisu -div free_*.sst		< http://localhost:8081/cgi-bin/sisu_sqlite.cgi >	1242
1227	or		or as instructed by command sisu -F webrick	1243
1228	sisu -dv import free_*.sst debian_constitution_v1.2.sst debian_social_contract_v1.1.sst gpl2.fsf.sst		(9) query the database using the search form	1244
1229	(5) use sisu to start the webrick (httpd) server (if it has not already been started):		if there are other options in the dropdown menu select	1245
1230	sisu -W		document_samples_sisu_markup	1246
1231	(6) use sisu to create a search form (for use with the webrick server, and your sample documents)		and search for some text, e.g.:	1247
			aim OR project	1248
			Γ selecting the *index* radio button gives an index of results using the object numbers	1249
			Γ selecting the *text* radio button gives the content of the matched paragraphs with the match highlighted	1250
			(10) to start again with a new database	1251
			to start from scratch you can drop the database with the command	1252

1253	sisu -dv dropall	1271	38.2.6 start the webrick server	
1254	and go to step 3		sisu -W	1272
1255	to get to step 3 in one step with a single command			
1256	sisu -dv recreate		38.3 remote placement of output	1273
1257	continue subsequent steps		configuration is necessary	1274
1258	38.2 misc		scp	1275
1259	38.2.1 url for output files -u -U		sisu -vr[and processing flags] [filename/wildcard]	1276
1260	sisu -uv[and other flags] [filename/wildcard]		rsync	1277
1261	sisu -Uv [filename/wildcard]		sisu -vR[and processing flags] [filename/wildcard]	1278
1262	38.2.2 toggle screen color			
1263	sisu -cv[and processing flags] [filename/wildcard]			
1264	38.2.3 verbose mode			
1265	sisu -v[and processing flags] [filename/wildcard]			
1266	sisu -V[and processing flags] [filename/wildcard]			
1267	38.2.4 quiet mode			
1268	sisu -q[and processing flags] [filename/wildcard]			
1269	38.2.5 maintenance mode intermediate files kept -M			
1270	sisu -Mv[and other flags] [filename/wildcard]			

1279	39. Configuration Files	40. Markup	1293
1280	Sample provided, on untarring the source tarball:	See sample markup provided on	1294
1281	conf/sisu/v2/sisurc.yml	< http://www.sisudoc.org/ >	1295
1282	conf/sisu/v3/sisurc.yml	< http://www.jus.uio.no/sisu >	1296
1283	and on installation under:	< http://www.jus.uio.no/sisu_markup >	1297
1284	/etc/sisu/v2/sisurc.yml	< http://www.jus.uio.no/sisu/SiSU >	1298
1285	/etc/sisu/v3/sisurc.yml	in particular for each of the document output samples provided,	1299
1286	The following paths are searched:	the source document is provided as well	
1287	./_sisu/v2/sisurc.yml or ./_sisu/v3/sisurc.yml	< http://www.jus.uio.no/sisu/SiSU/examples.html >	1300
1288	./_sisu/sisurc.yml	on untarring the source tarball:	1301
1289	~/_sisu/v2/sisurc.yml or ~/_sisu/v3/sisurc.yml	data/doc/sisu/markup-samples	1302
1290	~/_sisu/sisurc.yml	or the same once source is installed (or sisu-examples) under:	1303
1291	/etc/sisu/v2/sisurc.yml /etc/sisu/v3/sisurc.yml	/usr/share/doc/sisu/markup-samples/	1304
1292	/etc/sisu/sisurc.yml	and if you have sisu-markup-samples installed, under	1305
		data/doc/sisu/markup-samples-non-free/	1306
		/usr/share/doc/sisu/markup-samples-non-free/	1307
		Some notes are contained within the man page, man sisu and	1308
		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 ma-	1309
		terial. 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	1310
		Headers @headername: provide information related to the docu-	1311

	ment, this may relate to	<code>/ { italicise word or sentence } /</code>	1328
1312	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	<code>/italicisedword/</code>	1329
		<code>/italicisedword/</code>	1330
1313	2. semantic information about the document including the dublin core		
1314	40.2 Font Face	40.2.3 Underscore	1331
		<code><u>underscore word or sentence</u></code>	1332
		<code>- { underscore word or sentence } -</code>	1333
		<code>_underscoreword_</code>	1334
1315	Defaults are set. You may change the face to: bold, italics, underscore, strikethrough, ...	40.2.4 Strikethrough	1335
1316	40.2.1 Bold	<code>strikethrough word or sentence</code>	1336
		<code>- { strikethrough word or sentence } -</code>	1337
1317	bold: [list of words that should be made bold within document]	<code>-strikeword-</code>	1338
1318	bold line	<code>-strikeword-</code>	1339
1319	<code>!_ bold line</code>		
1320	bold word or sentence		
1321	<code>! { bold word or sentence } !</code>		
1322	<code>* { bold word or sentence } *</code>		
1323	<code>*boldword* or !boldword!</code>		
1324	<code>boldword or !boldword!</code>		
1325	40.2.2 Italics	40.3 Endnotes	1340
1326	italics: [list of words that should be italicised within document]	There are two forms of markup for endnotes, they cannot be mixed within the same document	1341
1327	<i>italicise word or sentence</i>	<code>here¹⁰⁹</code>	1342
		1. preferred endnote markup	1343
		<code>here~ { this is an endnote } ~</code>	1344
		2. alternative markup equivalent, kept because it is possible to search and replace to get markup in existing texts such as Project Gutenberg	1345
		<code>here~^</code>	1346
		<hr/> ¹⁰⁹ this is an endnote	

1347 ^ ~ this is an endote

1348 40.4 Links

1349 <SiSU>

1350 { SiSU }<http://www.sisudoc.org>



1351

1352 {sisu.png }<http://www.sisudoc.org>



1353

1354 { tux.png 64x80 }image

1355 <SiSU> ¹¹⁰

1356 {~^ SiSU }<http://www.sisudoc.org>

1357 is equivalent to:

1358 { SiSU }<http://www.sisudoc.org> ~{ <http://www.sisudoc.org> }~

1359 the same can be done with an image:



1360

1361 {~^ sisu.png "SiSU" }<http://www.sisudoc.org>

1362 40.5 Number Titles

Set with the header @markup:

1363

40.6 Line operations

1364

Line Operations (marker placed at start of line)

1365

!_ bold line

1366

bold line

1367

_1 indent paragraph one level

1368

indent paragraph one level

1369

_2 indent paragraph two steps

1370

indent paragraph two steps

1371

_* bullet paragraph

1372

Γ bullet paragraph

1373

1374

number paragraph (see headers for numbering document headings)

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

1375

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

1376

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

1377

40.7 Tables

1378

Table markup sample

1379

¹¹⁰<<http://www.sisudoc.org>>

¹¹¹<<http://www.sisudoc.org>>

```

1380 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

```

1381 Alternative form of table markup

```

1382 {t~h}
      |Mon|Tue|Wed|Thu|Fri|Sat|Sun
0    |*  |*  |*  |*  |*  |*  |*
1    |*  |*  |*  |*  |   |   |
2    |-  |*  |*  |*  |*  |*  |
3    |-  |*  |*  |*  |*  |*  |*
4    |-  |   |   |*  |*  |*  |
5    |*  |*  |*  |*  |*  |*  |*

```

1383 40.8 Grouped Text

```

1384     5.times { puts 'Ruby' }
1385 code{
1386     5.times { puts 'Ruby' }
1387 }code
1388
1388 A Limerick
1389
1389 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{
```

1390

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
```

40.9 Composite Document

1391

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

1392

```
ii { filename.sst }
```

1393

```
ii { filename.ssi }
```

1394

1395 **41. Change Appearance**1396 **41.1 Skins**

1397 “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.

1398 There are a few examples provided, on untarring the source tarball:

1399 `conf/sisu/skin/doc/`

1400 `data/doc/sisu/markup-samples/samples/_sisu/skin/doc/`

1401 and on installation under:

1402 `/etc/sisu/skin/doc/`

1403 `/usr/share/doc/sisu-markup-samples/v2/samples/_sisu/-skin/doc`

1404 The following paths are searched:

1405 `./_sisu/skin`

1406 `~/_sisu/skin`

1407 `/etc/sisu/skin`

1408 Skins under the searched paths in a per document directory, a per directory directory, or a site directory, named:

1409 `doc` [may be specified individually in each document]

1410 `dir` [used if identifier part of name matches markup directory name]

1411 `site`

1412 It is usual to place all skins in the document directory, with symbolic

links as required from dir or site directories.

41.2 CSS

1413

The appearance of html and XML related output can be changed for an output collection directory by preparing 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`

1415

`~/_sisu/css`

1416

and

1417

`/etc/sisu/css`

1418

The contents of the first directory found in the search path are copied to the corresponding sisu output directory with the command:

1419

`sisu -CC`

1420

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

1421

`dom.css html.css html.tables.css index.css sax.css xhtml.css`

1422

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

1423

css:

1424

[expand]

1425

1426 Extracts from the README

42. README 1427

SiSU 0.55 2007w27/6 2007-07-07 1428

Homepage: <<http://www.sisudoc.org>> 1429

old homepage: <<http://www.jus.uio.no/sisu>> 1430

Description 1431

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. 1432

With minimal preparation of a plain-text (UTF-8) file using its native markup-syntax, SiSU produces: plain-text, HTML, XHTML, EPUB, 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. 1433

Outputs share a common citation numbering system, and any semantic meta-data provided about the document. 1434

SiSU also provides concordance files, document content certificates and manifests of generated output. 1435

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). 1436

A vim syntax highlighting file and an ftplugin with folds for sisu markup is provided. Vim 7 includes syntax highlighting for SiSU 1437

.

1438 man pages, and interactive help are provided.

1439 Dependencies for various features are taken care of in sisu related packages. The package sisu-complete installs the whole of **SiSU**.

1440 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.

1441 Homepage: <<http://www.sisudoc.org>>

1442 old homepage: <<http://www.jus.uio.no/sisu>>

1443 **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.

1444 **SiSU** can be used for Internet, Intranet, local filesystem or cd publishing.

1445 **SiSU** can be used directly off the filesystem, or from a database.

1446 **SiSU**'s scalability, is be dependent on your hardware, and filesystem (in my case Reiserfs), and/or database Postgresql.

1447 Amongst it's characteristics are:

1448 Γ simple mnemonic markup style,

1449 Γ 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,

1450 Γ 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.

Γ it is command line driven, and can be set up on a remote server 1451

Γ 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. 1452

Input files should be UTF-8 1453

Once set up it is simple to use. 1454

42.1 Online Information, places to look 1455

<<http://www.sisudoc.org>> 1456

<<http://www.jus.uio.no/sisu>> 1457

Download Sources: 1458

<<http://www.jus.uio.no/sisu/SiSU/download.html#current>> 1459

<<http://www.jus.uio.no/sisu/SiSU/download.html#debian>> 1460

42.2 Installation 1461

NB. Platform is Unix / Linux. 1462

42.2.1 Debian 1463

If you use **Debian** use the **Debian** packages, check the information at: 1464

	<http://www.jus.uio.no/sisu/SiSU/download.html#debian>	1465	<http://www.jus.uio.no/sisu/SiSU/download.html#current>	1482
1466	(A) SiSU is available directly off the Debian archives for Sid and testing. It should necessary only to run as root:		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,	1483
1467	aptitude update			
1468	aptitude install sisu-complete		Ruby is the essential dependency for the basic operation of SiSU	1484
1469	(B) If there are newer versions of SiSU upstream of the Debian archives, they will be available by adding the following to your <code>/etc/apt/sources.list</code>		1. Download the latest source (information available) from:	1485
1470	deb <http://www.jus.uio.no/sisu/archive> unstable main non-free		<http://www.jus.uio.no/sisu/SiSU/download.html#current>	1486
1471	deb-src <http://www.jus.uio.no/sisu/archive> unstable main non-free		2. Unpack the source	1487
1472	[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]		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.	1488
1473	Then as root run:		42.2.4 to use setup.rb	1489
1474	aptitude update		this is a three step process, in the root directory of the unpacked SiSU as root type:	1490
1475	aptitude install sisu-complete		ruby setup.rb config	1491
1476	42.2.2 RPM		ruby setup.rb setup	1492
1477	RPMs are provided though untested, they are prepared by running alien against the source package, and against the debs.		as root:	1493
1478	They may be downloaded from:		ruby setup.rb install	1494
1479	<http://www.jus.uio.no/sisu/SiSU/download.html#rpm>		further information:	1495
1480	42.2.3 Source package .tgz		"><http://i.loveruby.net/en/projects/setup/>	1496
1481	Otherwise to install SiSU from source, check information at:		<http://i.loveruby.net/en/projects/setup/doc/usage.html>	1497
			42.2.5 to use install (prpared with "Rake")	1498
			Rake must be installed on your system:	1499

	<code><http://rake.rubyforge.org/></code>	1500	<code><http://rubyforge.org/frs/?group_id=615></code>	1519
1501	<code><http://rubyforge.org/frs/?group_id=50></code>		For a list of alternative actions you may type:	1520
1502	in the root directory of the unpacked SiSU as root type:		<code>ruby ./sisu-install help</code>	1521
1503	<code>rake</code>		<code>ruby ./sisu-install -T</code>	1522
1504	OR			
1505	<code>rake base</code>			
1506	This makes use of Rake (by Jim Weirich) and the provided Rakefile			
1507	For a list of alternative actions you may type:			
1508	<code>rake help</code>			
1509	<code>rake -T</code>			
1510	42.2.6 to use install (prpared with “Rant”)			
1511	(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:)			
1512	in the root directory of the unpacked SiSU as root type:			
1513	<code>ruby ./sisu-install</code>			
1514	OR			
1515	<code>ruby ./sisu-install base</code>			
1516	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.			
1517	further information:			
1518	<code><http://make.rubyforge.org/></code>			
			42.3 Dependencies	1523
			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.	1524
			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).	1525
			The following is from the debian/control file of sisu-3.0.2, which amongst other things provides the dependencies of sisu within Debian .	1526
			Source: sisu	1527
			Section: text	
			Priority: optional	
			Maintainer: SiSU Project < sisu@lists.sisudoc.org >	
			Uploaders: Ralph Amissah < ralph@amissah.com >	
			Build-Depends: debhelper (>= 8)	
			Standards-Version: 3.9.1	
			Homepage: http://www.sisudoc.org/	
			Vcs-Browser: http://git.sisudoc.org/?p=code/sisu.git	
			Vcs-Git: git://git.sisudoc.org/git/code/sisu.git	

```

XS-Dm-Upload-Allowed: yes

1528 Package: sisu
Architecture: all
Depends: ${misc:Depends}, ruby (>= 1.8.2),
libwebrick-ruby, rsync, unzip, zip
Recommends:
sisu-pdf, sisu-sqlite, sisu-postgresql, imagemagick,
keychain, librmagick-ruby,
librexml-ruby, openssl, openssh-client | lsh-client, tidy,
vim-addon-manager
Suggests: lv, calibre, pinfo, texinfo, trang
Conflicts: sisu-markup-samples (<= 1.0.11)
Replaces: sisu-markup-samples (<= 1.0.11)
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, EPUB, 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.

Package: sisu-complete
Architecture: all
Depends:
${misc:Depends}, ruby (>= 1.8.2), sisu (=
${source:Version}),

```

1529

```
sisu-pdf (= ${source:Version}), sisu-postgresql (=
${source:Version}),
```

```
sisu-sqlite (= ${source:Version})
Description: installs all SiSU related packages
SiSU is a lightweight markup based document structuring,
publishing and search
framework for document collections.
```

```
.
This package installs SiSU and related packages that
enable sisu to produce
pdf and to populate PostgreSQL and sqlite
databases.
```

```
.
See sisu for a description of the package.
```

```
1530 Package: sisu-pdf
Architecture: all
Depends:
${misc:Depends}, sisu, texlive-latex-base,
texlive-fonts-recommended,
texlive-latex-recommended, texlive-latex-extra,
texlive-xetex, lmodern,
ttf-liberation
Suggests: evince | pdf-viewer
Description: dependencies to convert SiSU LaTeX output to
pdf
SiSU is a lightweight markup based document structuring,
publishing and search
framework for document collections.
```

```
.
This package enables the conversion of SiSU LaTeX output
to pdf.
```

```
1531 Package: sisu-pdf
Architecture: all
Depends:
${misc:Depends}, sisu, texlive-latex-base,
```

```
texlive-fonts-recommended,
texlive-latex-recommended, texlive-latex-extra,
texlive-xetex, lmodern,
ttf-liberation
Suggests: evince | pdf-viewer
Description: dependencies to convert SiSU LaTeX output to
pdf
SiSU is a lightweight markup based document structuring,
publishing and search
framework for document collections.
```

```
.
This package enables the conversion of SiSU LaTeX output
to pdf.

Package: sisu-postgresql
Architecture: all
Depends:
${misc:Depends}, sisu, libdbd-pg-ruby, libdbd-pg-ruby1.8,
libdbi-ruby,
libdbi-ruby1.8, libdbm-ruby, postgresql, libfcgi-ruby1.8 |
libfcgi-ruby1.9.1
Suggests: postgresql-contrib
Description: SiSU dependencies for use with PostgreSQL
database
SiSU is a lightweight markup based document structuring,
publishing and search
framework for document collections.
```

1532

```
.
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.

1533 Package: sisu-sqlite
 Architecture: all
 Depends:
 \${misc:Depends}, sisu, sqlite3, libsqlite3-ruby,
 libdbd-sqlite3-ruby,
 libdbd-sqlite3-ruby1.8, libdbi-ruby, libdbi-ruby1.8,
 libdbm-ruby,
 libfcgi-ruby1.8 | libfcgi-ruby1.9.1
 Description: SiSU dependencies for use with SQLite
 database
 SiSU is a lightweight markup based document structuring,
 publishing and search
 framework for document collections.
 .
 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.

1534 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). 1535

After installation of sisu-complete, move to the document samples
 directory 1536

```
cd /usr/share/doc/sisu/markup-samples/samples 1537
```

and run 1538

```
sisu -3 free_as_in_freedom.rms_and_free_software.sam_williams.sst 1539
```

or the same: 1540

```
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 1542

or to generate an online document move to a writable directory, as
 the file will be downloaded there and e.g. 1543

```
sisu -3 <http://www.jus.uio.no/sisu/src/free\_culture.lawrence\_lessig.sst> 1544
```

the database stuff is extra perhaps, the latex stuff could be consid-
 ered extra perhaps but neither needs to be installed for most of sisu
 output to work 1545

examine source document, vim has syntax support 1546

```
gvim free_as_in_freedom.rms_and_free_software.sam_williams.sst 1547
```

additional markup samples in 1548

```
<http://www.jus.uio.no/sisu/SiSU/examples.html> 1549
```

For help 1550

```
man sisu 1551
```

or 1552

1553	sisu -help	There are post installation steps (which are really part of the overall installation)	1570
1554	e.g.		
1555	for the way sisu “sees/maps” your system	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).	1571
1556	sisu -help env		
1557	for list of commands and so on		
1558	sisu -help commands		
1559	42.5 Configuration files	42.6 Use General Overview	1572
1560	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	Documents are marked up in SiSU syntax and kept in an ordinary text editable file, named with the suffix .sst, or .ssm	1573
1561	Γ configuration file - a yaml file	Marked up SiSU documents are usually kept in a sub-directory of your choosing	1574
1562	/etc/sisu/[sisu version]/sisurc.yml	use the interactive help and man pages	1575
1563	~/.sisu/sisurc.yml	sisu -help	1576
1564	Γ directory structure - setting up of output and working directory.	man sisu	1577
1565	* skins - changing the appearance of a project, directory or individual documents within ~/.sisu/skin	42.7 Help	1578
1566	~/.sisu/skin/doc contains individual skins, with symbolic links from	interactive help described below, or man page:	1579
1567	~/.sisu/skin/dir if the contents of a directory are to take a particular document skin.	man sisu	1580
1568	Γ 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.	man 8 sisu	1581
1569	Γ if you use Vim as editor there is a syntax highlighter and fold resource config file for SiSU . I hope more syntax highlighters follow.	‘man sisu_markup-samples’ [if the sisu-markup-samples package is also installed]	1582
		Once installed an interactive help is available typing ‘sisu’ (without) any flags, and select an option:	1583
		sisu	1584
		alternatively, you could type e.g.	1585
		sisu -help commands	1586

1587	sisu -help env	1607	Once installed there is some information on SiSU Markup in its help:	
1588	sisu -help headers		sisu -help markup	1608
1589	sisu -help markup		and	1609
1590	sisu -help headings		sisu -help headers	1610
1591	etc.		Sample marked up document are provided with the download tarball in the directory:	1611
1592	for questions about mappings, output paths etc.		./data/doc/sisu/markup-samples/	1612
1593	sisu -help env		These are installed on the system usually at:	1613
1594	sisu -help path		/usr/share/doc/sisu/markup-samples/	1614
1595	sisu -help directory		More markup samples are available in the package sisu-markup-samples, which if installed should be available at:	1615
1596	42.8 Directory Structure		/usr/share/doc/sisu/markup-samples-non-free/	1616
1597	Once installed, type:		Many more are available online off:	1617
1598	sisu -help env		< http://www.jus.uio.no/sisu/SiSU/examples.html >	1618
1599	OR			
1600	sisu -V			
1601	42.9 Configuration File		42.11 Additional Things	1619
1602	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)		There is syntax support for some editors provided (together with a README file) in	1620
1603	42.10 Markup		./data/sisu/v2/conf/editor-syntax-etc	1621
1604	See man pages.		usually installed to:	1622
1605	man sisu		/usr/share/sisu/v2/conf/editor-syntax-etc	1623
1606	man 8 sisu		42.12 License	1624
			License: GPL 3 or later see the copyright file in	1625
			./data/doc/sisu	1626

usually installed to:

1627

1628 /usr/share/doc/sisu

1629 **42.13 SiSU Standard**

1630 **SiSU** uses:

1631 Γ Standard **SiSU** markup syntax,

1632 Γ Standard **SiSU** meta-markup syntax, and the

1633 Γ Standard **SiSU** object citation numbering and system

1634 © Ralph Amissah 1997, current 2006 All Rights Reserved.

1635 Γ however note the License section

1636 CHANGELOG

1637 ./CHANGELOG

1638 and see

1639 <<http://www.jus.uio.no/sisu/SiSU/changelog.html>>

1640 <http://www.jus.uio.no/sisu/SiSU/changelog_markup_samples.html>

1641 Extracts from man 8 sisu

43. Post Installation Setup 1642

43.1 Post Installation Setup - Quick start 1643

After installation of sisu-complete, move to the document samples directory,

```
cd /usr/share/doc/sisu/markup-samples/samples 1645
```

[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] 1646

and in that directory, initialise the output directory with the command 1647

```
sisu -CC 1648
```

then run: 1649

```
sisu -1 free_as_in_freedom.rms_and_free_software.sam_williams.sst 1650
```

or the same: 1651

```
sisu -NhwpoabxXyv free_as_in_freedom.rms_and_free_software.sam_williams.sst 1652
```

look at output results, see the “sisu_manifest” page created for the document 1653

for an overview of your current sisu setup, type: 1654

```
sisu -help env 1655
```

or 1656

```
sisu -V 1657
```

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. 1658

```
sisu -1 <http://www.jus.uio.no/sisu/src/gpl.fsf.sst> 1659
```

```
sisu -3 <http://www.jus.uio.no/sisu/src/free_culture.lawrence_lessig.sst> 1660
```

1661	examine source document, vim has syntax highlighting support		
1662	gvim free_as_in_freedom.rms_and_free_software.sam_williams.sst		
1663	additional markup samples in		
1664	<code><http://www.jus.uio.no/sisu/SiSU/examples.html></code>		
1665	it should also be possible to run sisu against sisupods (prepared zip files, created by running the command <code>sisu -S [filename]</code>), whether stored locally or remotely.		
1666	<code>sisu -3 <http://www.jus.uio.no/sisu/pod/free_culture.lawrence_lessig.sst.zip></code>		
1667	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:		
1668	<code>sisu -3 --trust <http://www.jus.uio.no/sisu/pod/free_culture.lawrence_lessig.sst.zip></code>		
1669	For help		
1670	<code>man sisu</code>		
1671	<code>sisu --help</code>		
1672	<code>sisu --help env</code> for the way sisu “sees/maps” your system		
1673	<code>sisu --help commands</code> for list of commands and so on		
1674	43.2 Document markup directory		
1675	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:		
1676	<code>mkdir ~/sisu_test</code>		
		<code>cd ~/sisu_test</code>	1677
		<code>cp -a /usr/share/doc/sisu/markup-samples/samples/* ~/sisu_test/.</code>	1678
		better if you have installed sisu-markup-samples	1679
		<code>cp -a /usr/share/doc/sisu/markup-samples-non-free/samples/* ~/sisu_test/.</code>	1680
		Tip:	1681
		<code>sisu -U [sisu markup filename]</code>	1682
		should printout the different possible outputs and where sisu would place them.	1683
		Tip: if you want to toggle ansi color add	1684
		<code>c</code>	1685
		to your flags.	1686
		43.2.1 Configuration files	1687
		SiSU configuration file search path is:	1688
		<code>./_sisu/sisurc.yaml</code>	1689
		<code>~/_sisu/sisurc.yaml</code>	1690
		<code>/etc/sisu/sisurc.yaml</code>	1691
		<code>./%% Debian Installation Note</code>	1692
		43.2.2 Debian INSTALLATION Note	1693
		It is best you see	1694
		<code><http://www.jus.uio.no/sisu/SiSU/download.html#debian></code>	1695

	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) :	1696	packages. The package <code>sisu-complete</code> installs the whole of SiSU
1698	Package: <code>sisu</code>	1697	.
1699	SiSU is a lightweight markup based, command line oriented, document structuring, publishing and search framework for document collections.		Additional document markup samples are provided in the package <code>sisu-markup-samples</code> 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. 1705
1700	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), EPUB, 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.		Homepage: < http://www.sisudoc.org > 1706 old homepage: < http://www.jus.uio.no/sisu > 1707
1701	SiSU also provides concordance files, document content certificates and manifests of generated output.		43.2.3 Document Resource Configuration 1708
1702	A vim syntax highlighting file and an ftplugin with folds for <code>sisu</code> markup is provided, as are syntax highlighting files for <code>kate</code> , <code>kwrite</code> , <code>gedit</code> and <code>diakonos</code> . Vim 7 includes syntax highlighting for SiSU .		sisu resource configuration information is obtained from sources (where they exist): 1709
1703	man pages, and interactive help are provided.		~/.sisu/sisurc.yaml 1710
1704	Dependencies for various features are taken care of in <code>sisu</code> related		/etc/sisu/[sisu version]/sisurc.yaml 1711
			<code>sisu</code> program defaults 1712
			43.2.4 Skins 1713
			Skins default document appearance may be modified using skins contained in sub-directories located at the following paths: 1714
			./_sisu/skin 1715
			~/.sisu/skin 1716
			/etc/sisu/skin 1717
			more specifically, the following locations (or their <code>/etc/sisu</code> equivalent) should be used: 1718
			~/.sisu/skin/doc 1719
			skins for individual documents; 1720
			~/.sisu/skin/dir 1721

1722 skins for directories of matching names;
 1723 `~/sisu/skin/site`
 1724 site-wide skin modifying the site-wide appearance of docu-
 ments.
 1725 Usually all skin files are placed in the document skin direc-
 tory:
 1726 `~/sisu/skin/doc`
 1727 with softlinks being made to the skins contained there from other
 skin directories as required.

44. FAQ - Frequently Asked/Answered Questions 1728

44.1 Why are urls produced with the `-v` (and `-u`) flag that point to a web server on port 8081 ? 1729

Try the following rune: 1730

```
Γ sisu -W 1731
```

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. 1732

44.2 I cannot find my output, where is it? 1733

The following should provide help on output paths: 1734

```
Γ sisu -help env 1735
```

```
Γ sisu -V [same as the previous command] 1736
```

```
Γ sisu -help directory 1737
```

```
Γ sisu -help path 1738
```

```
Γ sisu -U [filename] 1739
```

```
Γ man sisu 1740
```

44.3 I do not get any pdf output, why? 1741

SiSU produces LaTeX and pdflatex is run against that to generate pdf files. 1742

If you use **Debian** the following will install the required dependencies 1743

```
Γ aptitude install sisu-pdf 1744
```

the following packages are required: tetex-bin, tetex-extra, latex-ucs 1745

1746 **44.4 Where is the latex (or some other interim)
output?**

1747 Try adding -M (for maintenance) to your command flags, e.g.:

1748 `Γ sisu -HpMv [filename]`

1749 this should result in the interim processing output being retained,
and information being provided on where to find it.

1750 `Γ sisu -help directory`

1751 `Γ sisu -help path`

1752 should also provide some relevant information as to where it is
placed.

1753 **44.5 Why isn't SiSU markup XML**

1754 I worked with text and (though I find XML immensely valuable)
disliked noise ... better to sidestep the question and say:

1755 **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

1756 `sisu -to-sax [filename/wildcard]`

1757 `sisu -to-dom [filename/wildcard]`

1758 `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]` 1760

**44.6 LaTeX claims to be a document preparation system
for high-quality typesetting. Can the same be said
about SiSU?** 1761

SiSU is not really about type-setting. 1762

LaTeX is the ultimate computer instruction type-setting language
for paper based publication. 1763

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. 1764

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. 1765

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 1766

things with different needs for layout - as indeed is what is needed to store information in a database in searchable objects.

1767 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.

1768 Where there are large document sets, it provides consistency in appearance in each output format for the documents.

1769 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 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.

1770 **44.7 Can the SiSU markup be used to prepare for a LaTeX automatic building of an index to the work?**

1771 Has not been, is of interest though the question on introducing such possibilities is how to keep them as unobtrusive as possible, and as generically relevant as possible to other output formats (which is why the focus on object numbers). Unobtrusive refers both to the markup (where there is no big problem with introducing optional extras); and, more challengingly how to minimise impact on competing ideas/interests, such allowing the addition of semantic tags

which could be tied to objects, mapped against the objects that contain them, (permitting mapping and mining of content in various ways that would be largely agnostic of output format - object numbering being an attempt to move beyond output format based content locators (such as page numbers). The desire being to (be a meta markup and) maintain agnosticism as to what is being generated and in development to favor solutions of that nature. Keep bridging LaTeX, XML, SQL ... make use of objects and serialisation for mapping whether against content or meta-content (such as semantic [or additional structural] markers).

44.8 Can the conversion from SiSU to LaTeX be modified if we have special needs for the LaTeX, or do we need to modify the LaTeX manually? 1772

Should be possible to modify code, it is GPLv3, should be possible either to modify existing modules or write an independent module for generating bespoke latex. Generic improvements are welcome for inclusion/incorporation in the existing code base. 1773

If there are tools to generate mathematical/scientific formula from latex to images (jpg, png), the latex parser could conceivably be used to make these available to other output formats. 1774

44.9 How do I create GIN or GiST index in Postgresql for use in SiSU 1775

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: 1776

“I have tsearch2 slides which introduces tsearch2 <<http://www.sai.msu.su/~megera/wiki/tsearch2slides>> 1777

1778 FTS in PostgreSQL is provided by tsearch2, which should work without any indices (GiST or GIN) ! Indices provide performance, not functionality.

1779 In your example I'd do (simple way, just for demo):

1780 **0.** compile, install tsearch2 and load tsearch2 into your database

1781 `cd contrib/tsearch2; make&&make&&install&&make installcheck; psql DB ; tsearch2.sql`

1782 **1.** Add column fts, which holds tsvector

1783 `alter table documents add column fts tsvector;`

1784 **2.** Fill fts column

1785 `update document set fts = to_tsvector(clean);`

1786 **3.** create index - just for performance !

1787 `create index fts_gin_idx on document using gin(fts);`

1788 **4.** Run vacuum

1789 `vacuum analyze document;`

1790 That's all.

1791 Now you can search:

1792 `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;`

1793 **44.10 Are there some examples of using Ferret Search with a SiSU repository?**

1794 Heard good things about Ferret, but have not used it. The output directory structure and content produced by **SiSU** is very uniform. Have looked at a couple of other engines (hyperrestrai, lucene).

There it was enough to identify the files that needed to be indexed and pass them to the search indexing tool. Some Unix runde doing the job, such as:

```
1795 find /home/ralph/sisu_www -type f | \
    egrep '/sisu_www/(sisu|document_archive)/.+?.html$' | \
1796 \
    egrep -v '(doc|concordance).html$' | \
    estcmd gather -sd casket -
```

you would have to experiment with what gives the desired result, the file doc.html is the complete text in html (there are additional smaller html segments), and plain.txt the document as a text file. It may be possible to index the text file and return the html document.

1797 **Have you had any reports of building SiSU from tar on Mac OS 10.4?**

1798 None. In the early days of its release a Mac friend built and run the ruby code part that did not rely on system calls to bits like the latex engine. That is already some years back. He was not into writing or document markup, and did it as a favour at the time. I have not followed up that thread of development.

1799 It should however be possible, much of the output relies on plain ruby, and the system commands to latex etc. could be made appropriate for the underlying OS.

1800 **44.12 Where is version 1?**

1801 Version 1 was finally released on December 21, 2009, and largely to make a version 1 branch available as version 2 was in the pipeline with an imminent release. Most of **SiSU** was mature and stable long before the release of version 1.

1802 **44.13 What is the difference between version 1 and
2?**

1803 Input and output of version 1 and 2 are largely the same with the following significant exceptions. On the input side, document headers, that is metadata and processing instructions have changed in version 2. On the output side, version 2 was introduced with EPUB documents as a possible output, and over time is likely to be developed further.

1804 Version 2 introduces a new processing layer, which relies more on the programming language **Ruby** 's objects (and regular expressions), than version 1 does which relies on regular expressions. The thinking behind version 1 use of regular expressions was that it made it more straightforward to switch languages for processing, as many languages support regular expressions; the thinking behind version 2 was version 1 was more complicated than it need be, and since **Ruby** was the language used, why not make programming more straightforward, as it would be easier to develop further.

1805 Version 1 was removed with the introduction of the version 3 development branch

45. Who might be interested in the SiSU feature set?

1806

SiSU is most likely to be of interest to people who are working with medium to large volumes of published texts that would like to have the presented in a uniform way that is searchable (either using sisu database integration or an appropriate indexing tool), with the possibility of multiple alternative output formats that may be added to and upgraded/updated over time. **SiSU** should be of interest to institutions/ organisations/ governments/ individuals with document collections and some technical knowhow that are interested in:

1807

Γ long term maintenance and reducing downstream/future costs of maintaining those document sets for which **SiSU** is suited.

1808

Γ the ability to output multiple standard format outputs for various purposes.

1809

Γ the implications for search offered

1810

1811 **46. Work Needed**

1812 **SiSU** is fairly mature and for most purposes the syntax and what it is supposed to do is clear. For the most part additions and changes are minor and backward compatible, (in particular there may be things of interest that to be able to achieve will require additions to the syntax).

1813 Γ Amongst the most requested features is a way to represent and extract bibliographies from scholarly and other writings. This involves an extension of sisu markup syntax and a new module to extract the bibliography.

1814 Γ Integration of postgresql tsearch2 / gin indexing, (which currently needs to be done manually, and) which has been waiting for the integration of tsearch2 / gin into Postgresql main, which is supposed to occur in Postgresql 8.3

1815 Γ Internationalisation always. **SiSU** is utf-8 and for those parts that are utf-8 friendly will work out of the box - html and postgresql for example work out of the box (and for example comfortably represent Chinese text), LaTeX and odf do not work out of the box, they need additional work for extended language sets.

1816 Γ Refinements and improvements to output representations, some are fairly mature, others (such as manpages and info files (and even ODF) remain young.

1817 Γ Simple extension to contain, link and share included audio and multi-media files, (including sisupod.zip)

47. Wishlist 1818

SiSU provides a lot of “plumbing” and is readily usable as a tool by those comfortable with marking up documents with an editor. The syntax is fairly easy to learn, especially the subset required to start using **SiSU** effectively. 1819

SiSU might also be of interest to developers interested in: 1820

Γ experimenting with the search implications offered 1821

Γ producing additional output formats 1822

Γ producing conversion tools 1823

Γ producing input interfaces, (experimenting with additional interfaces for producing sisu source documents) 1824

Several tools that are of interest would come under the heading interface and conversion. Amongst others, the following are of interest: 1825

Γ Converters from various document formats, such as Open Document Text (ODF), MS Word(TM) and Word Perfect(TM), even html. The problem here is one of the most important things for **SiSU** is to be able to recognise the structure of a document, and many documents prepared in other formats have not been prepared strictly with a view to representing structure, but appearance - so heading levels may be “painted” to look right rather than have the correct structural representation. Even if conversion is not perfect this may serve as a first step in assisting in conversion of documents to **SiSU** for those with legacy document sets that they would like to have in sisu format. (once in **SiSU** it is easier to get out in various other formats as this is what sisu does, within the constraints of the information that sisu uses to generate output) 1826

Γ The possibility to save directly from from various word proces- 1827

sors, and possibly templates within them to assist in making sure the document structure is “understood” by SiSU .

- 1828 Γ Web interface/front-end, a form like front end for the writing or submission of sisu documents to a server which uses SiSU to generate output. Headers could be made available as separate small entry forms with help provided to explain where they might be used. Apart from the most important headers such as title, author, date and possibly subject the remainder of the header forms could be placed after the form for substantive content. This would offer a more Web 2.0 like approach to the use of SiSU and the possibility of using it for collaborative editing of content (possibly for documents that are to be finalised/published as the citation system is most suited to published works). [Collaborative editing is currently possible through use of a collaborative editor such as Gobby which makes use of the Obby protocol].

48. Editor Files, Syntax Highlighting 1829

The directory: 1830

./data/sisu/v2/conf/editor-syntax-etc/ 1831

./data/sisu/v3/conf/editor-syntax-etc/ 1832

/usr/share/sisu/v2/conf/editor-syntax-etc 1833

/usr/share/sisu/v3/conf/editor-syntax-etc 1834

contains rudimentary sisu syntax highlighting files for: 1835

Γ (g)vim <<http://www.vim.org>> 1836

package: sisu-vim 1837

status: largely done 1838

there is a vim syntax highlighting and folds component 1839

Γ gedit <<http://www.gnome.org/projects/gedit>> 1840

Γ gobby <<http://gobby.0x539.de/>> 1841

file: sisu.lang 1842

place in: 1843

/usr/share/gtksourceview-1.0/language-specs 1844

or 1845

~/gnome2/gtksourceview-1.0/language-specs 1846

status: very basic syntax highlighting 1847

comments: this editor features display line wrap and is used by Goby! 1848

Γ nano <<http://www.nano-editor.org>> 1849

file: nanorc 1850

save as: 1851

~/nanorc 1852

	status: basic syntax highlighting	18853	to file ~/.emacs add the following 2 lines:	
1854	comments: assumes dark background; no display line-wrap; does line breaks		(add-to-list 'load-path “/usr/share/sisu/v2/conf/editor- syntax-etc/emacs”)	1878
1855	Γ diakonos (an editor written in ruby) < http://purepistos.net/ diakonos >		(require 'sisu-mode.el)	1879
1856	file: diakonos.conf		[not done / not yet included]	1880
1857	save as:		Γ vim & gvim < http://www.vim.org >	1881
1858	~/diakonos/diakonos.conf		files:	1882
1859	includes:		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)	1883
1860	status: basic syntax highlighting		status: this includes: syntax highlighting; vim folds; some error checking	1884
1861	comments: assumes dark background; no display line-wrap		comments: this editor features display line wrap	1885
1862	Γ kate & kwrite < http://kate.kde.org >		NOTE:	1886
1863	file: sisu.xml		[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]	1887
1864	place in:			
1865	/usr/share/apps/katepart/syntax			
1866	or			
1867	~/kde/share/apps/katepart/syntax			
1868	[settings::configure kate::{highlighting,filetypes}]			
1869	[tools::highlighting::<markup,scripts>]			
1870	Γ nedit < http://www.nedit.org >			
1871	file: sisu_nedit.pats			
1872	nedit -import sisu_nedit.pats			
1873	status: a very clumsy first attempt [not really done]			
1874	comments: this editor features display line wrap			
1875	Γ emacs < http://www.gnu.org/software/emacs/emacs.html >			
1876	files: sisu-mode.el			

1888	49. Help Sources	1908	49.2 sisu generated output - links to html	
1889	49.1 man pages	1909	Note SiSU documentation is prepared in SiSU and output is available in multiple formats including amongst others html, pdf, odf and epub, which may be also be accessed via the html pages ¹¹²	
1890	man sisu			
1891	man sisu-concordance			
1892	man sisu-epub		49.2.1 www.sisudoc.org	1910
1893	man sisu-git		< http://sisudoc.org/sisu/sisu_manual/index.html >	1911
1894	man sisu-harvest		< http://sisudoc.org/sisu/sisu_manual/index.html >	1912
1895	man sisu-html		49.3 man2html	1913
1896	man sisu-odf		49.3.1 locally installed	1914
1897	man sisu-pdf		file:///usr/share/doc/sisu/html/sisu.1.html	1915
1898	man sisu-pg		file:///usr/share/doc/sisu/html/sisu.1.html	1916
1899	man sisu-po		/usr/share/doc/sisu/html/sisu_pdf.7.html	1917
1900	man sisu-sqlite		/usr/share/doc/sisu/html/sisu_postgresql.7.html	1918
1901	man sisu-txt		/usr/share/doc/sisu/html/sisu_sqlite.7.html	1919
1902	man 7 sisu_complete		/usr/share/doc/sisu/html/sisu_webrick.1.html	1920
1903	man 7 sisu_pdf		49.3.2 www.jus.uio.no/sisu	1921
1904	man 7 sisu_postgresql		< http://www.jus.uio.no/sisu/man/sisu.1.html >	1922
1905	man 7 sisu_sqlite		< http://www.jus.uio.no/sisu/man/sisu.1.html >	1923
1906	man sisu_termsheet		< http://www.jus.uio.no/sisu/man/sisu_complete.7.html >	1924
1907	man sisu_webrick		< http://www.jus.uio.no/sisu/man/sisu_pdf.7.html >	1925

¹¹²named index.html or more extensively through sisu_manifest.html

- 1926 <http://www.jus.uio.no/sisu/man/sisu_postgresql.7.html>
- 1927 <http://www.jus.uio.no/sisu/man/sisu_sqlite.7.html>
- 1928 <http://www.jus.uio.no/sisu/man/sisu_webrick.1.html>

Metadata

SiSU Metadata, document information

Document Manifest @:

<http://www.sisudoc.org/sisu/sisu_manual/sisu_manifest.html>

Title: SiSU - Manual

Creator: Ralph Amissah

Rights: Copyright (C) Ralph Amissah 2007, part of SiSU documentation, License GPL 3;

Subject: ebook, epubliishing, electronic book, electronic publishing, electronic document, electronic citation, data structure, citation systems, search

Publisher: SiSU <<text:a xlink:type='simple' xlink:href='http://www.jus.uio.no/sisu'62;http://www.jus.uio.no/sisu#60;/text:a#62;> (this copy)

Date created: 2002-08-28

Date issued: 2002-08-28

Date available: 2002-08-28

Date modified: 2011-02-07

Date: 2008-07-21

Topics Registered: SiSU>manual;electronic documents:SiSU>manual

Version Information

Sourcefile: sisu_manual.ssm.sst

Filetype: SiSU text insert 2.0

Source Digest: SHA256(sisu_manual.ssm.sst)=6e9af9f4f6be91b82bddcaa3-ed90e80f9a4a25d878cf4a64c168e41f223e246d

Skin Digest: SHA256(skin_sisu_manual.rb)=689f79b53be3d51460af31dc82d6cbab-4a253079db263cd715c8254a6af60eb8

Generated

Document (dal) last generated: Fri Mar 11 23:48:55 +0100 2011

Generated by: SiSU 2.8.2 of 2011w10/5 (2011-03-11)

Ruby version: ruby 1.8.7 (2008-08-11 patchlevel 72) [i486-linux]