SiSU
Search

Ralph Amissah

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

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

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

Contents

1. SiSU Search - Introduction

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

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

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

2. SQL

2.1 populating SQL type databases

SiSU feeds sisu markupd documents into sql type databases PostgreSQL[1] and/or SQLite[2] database together with information related to document structure.

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

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

There is of course the possibility to add further structures.

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

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

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

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

3. Postgresql

3.1 Name

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

3.2 Description

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

3.3 Synopsis

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

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

3.4 Commands

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

-D or –pgsql may be used interchangeably.

3.4.1 create and destroy database

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

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

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

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

sisu -D –recreate

destroys existing database and builds a new empty database structure

3.4.2 import and remove documents

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

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

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

4. Sqlite

4.1 Name

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

4.2 Description

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

4.3 Synopsis

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

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

4.4 Commands

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

-d or –sqlite may be used interchangeably.

4.4.1 create and destroy database

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

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

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

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

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

4.4.2 import and remove documents

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

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

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

5. Introduction

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

Sample search frontend [3] A small database and sample query front-end (search from) that makes use of the citation system, object citation numbering to demonstrates functionality.[4]

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

Note you may set results either for documents matched and object number locations within each matched document meeting the search criteria; or display the names of the documents matched along with the objects (paragraphs) that meet the search criteria.[5]

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

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

sisu –help sql

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

Note on databases built

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

5.2 Search Form

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

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

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

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

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

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

[4] (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.

[5] 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.

6. Hyperestraier

See the documentation for hyperestraier:

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

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

man estcmd

on sisu_hyperestraier:

man sisu_hyperestraier

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

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

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

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

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

cd /home/ralph/sisu_www

estcmd gather -sd casket /home/ralph/sisu_www

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

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

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

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

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

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

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

(B) to set up the search form

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

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

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

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

[see estraier documentation for paths]

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

indexname: /home/ralph/sisu_www/casket

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

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

(C) to test using webrick, start webrick:

sisu -W

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

DOCUMENT INFORMATION (METADATA)

Metadata

Document Manifest @

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

Dublin Core (DC)

DC tags included with this document are provided here.

DC Title: SiSU - Search

DC Creator: Ralph Amissah

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

DC Type: information

DC Date created: 2002-08-28

DC Date issued: 2002-08-28

DC Date available: 2002-08-28

DC Date modified: 2007-09-16

DC Date: 2007-09-16

Version Information

Sourcefile: sisu_search._sst

Filetype: SiSU text insert 0.58

Sourcefile Digest, MD5(sisu_search._sst)= c085c2eb6d68f1b7d50435f673ede407

Skin_Digest: MD5(/home/ralph/grotto/theatre/dbld/builds/sisu/sisu/data/doc/sisu/sisu_markup_samples/sisu_m
20fc43cf3eb6590bc3399a1aef65c5a9

Generated

Document (metaverse) last generated: Tue Sep 25 02:54:29 +0100 2007

Generated by: SiSU 0.59.1 of 2007w39/2 (2007-09-25)

Ruby version: ruby 1.8.6 (2007-06-07 patchlevel 36) [i486-linux]

Information on this document copy and an unofficial List of Some web related information and sources

”Support Open Standards and Software Libre for the Information Technology Infrastructure” RA

Information on this document copy www.jus.uio.no/sisu/

W3 since October 3 1993 SiSU 1997, current 2007.
SiSU presentations at **www.jus.uio.no/sisu/**

SiSU **pdf** versions can be found at:
http://www.jus.uio.no/sisu/sisu_search/portrait.pdf
http://www.jus.uio.no/sisu/sisu_search/landscape.pdf

SiSU **html** versions may be found at:
http://www.jus.uio.no/sisu/sisu_search/toc.html or
http://www.jus.uio.no/sisu/sisu_search/doc.html

SiSU Manifest of document output and metadata may be found at:
http://www.jus.uio.no/sisu/sisu_search/sisu_manifest.html

SiSU found at: **www.jus.uio.no/sisu/**

Links that may be of interest at SiSU and elsewhere:

SiSU Manual
http://www.jus.uio.no/sisu/sisu_manual/

Book Samples and Markup Examples
http://www.jus.uio.no/sisu/SiSU/2.html

SiSU @ Wikipedia
http://en.wikipedia.org/wiki/SiSU

SiSU @ Freshmeat
http://freshmeat.net/projects/sisu/

SiSU @ Ruby Application Archive
http://raa.ruby-lang.org/project/sisu/

SiSU @ Debian
http://packages.qa.debian.org/s/sisu.html

SiSU Download

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

SiSU Changelog

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

SiSU help

http://www.jus.uio.no/sisu/sisu_manual/sisu_help/

SiSU help sources

http://www.jus.uio.no/sisu/sisu_manual/sisu_help_sources/

SiSU home:
www.jus.uio.no/sisu/