

VIVO and the Solr search engine

- [What is Solr?](#)
- [How does VIVO use Solr?](#)
 - [Solr for the end user.](#)
 - [Solr within VIVO](#)
- [How is Solr created and configured?](#)
- [The search index](#)
 - [What is in the index?](#)
 - [What is in each record?](#)
 - [Customizing the index](#)
- [How does VIVO contact Solr?](#)

VIVO and Solr are two distinct web applications that act as one. Solr gives users the ability to search the VIVO data. VIVO also uses Solr for some of its internal data retrieval.

What is Solr?

Solr is an open-source, enterprise level search platform, available from Apache. It is based on the popular Lucene search engine. VIVO uses a standard instance of Solr, without modification. You can learn more about Solr at the [Apache Solr home page](#).

VIVO maintains its data in a semantic triple-store. A triple-store is very well suited for expressing a complex, flexible web of data relationship. It is not very well suited for text-based searches. Solr provides fast searching with features like

- weighted results by field,
- searching by the stems of words, rather than exact matches,
- faceted search results,
- and much more.

And Solr provides these features much more efficiently than a triple-store would.

Solr maintains its own index of data, which reflects the contents of the triple-store. As the data in VIVO changes, the contents of the Solr index must change also. In most cases this happens automatically, but not always. Sometimes the search index must be rebuilt to bring it into synchronization with the triple-store. See the section below called "[How is the index kept up to date](#)" for more information.

Solr is implemented as a self-contained web application, separate from VIVO. At most VIVO sites, Solr and VIVO run on the same machine, in the same instance of Tomcat, but this is not the only possible configuration. It is possible to put Solr in a different servlet container or even on a different computer from VIVO.

In a typical VIVO installation, Solr is hidden behind VIVO, and the users cannot access it directly. In general, they don't know that Solr exists as an application.

How does VIVO use Solr?

VIVO uses the Solr search engine in two ways:

- as a service to the end user,
- as a tool within the structure of the application.

Solr for the end user.

Like many web sites, VIVO includes a search box on every page. The person using VIVO can type a search term, and see the results. This search is conducted by Solr, and the results are formatted and displayed by VIVO.



Search results for 'oncology'

[Not the results you expected?](#)

[oncology](#)

[Radiation Oncology](#) | Clinical Section

[Gynecological Oncology](#) | Journal

... Gynecological **Oncology** Cyclooxygenase 1 and 2 mRNA and protein expression in the Gallus domesticus model of ovarian cancer Collection Information Resource ...

[Medical Oncology](#) | Clinical Section

[Translational Oncology](#) | Journal

... Translational **Oncology** Computed tomography assessment of response to therapy: Tumor volume change measurement, truth data, and error Collection Information ...

[Gynecologic Oncology](#) | Journal

... Gynecologic **Oncology** A phase II trial of interleukin-12 in patients with advanced cervical cancer: clinical and immunologic correlates. Eastern Cooperative ...

[Radiation Oncology](#) | Organization

... Assistant Professor of Radiation **Oncology** Assistant Professor of Clinical Radiation **Oncology** Assistant Professor of Clinical Radiation **Oncology** Professor ...

Display only

[people](#)

[activities](#)

[courses](#)

[events](#)

[organizations](#)

[research](#)

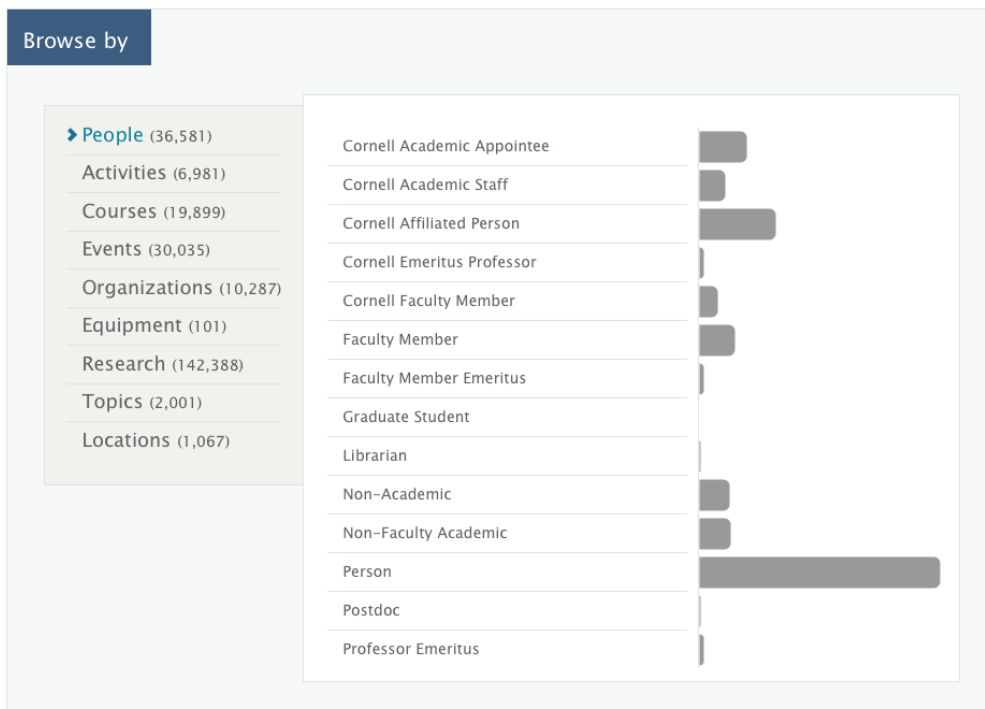
[topics](#)

Solr allows for a "faceted" search, and VIVO displays the facets on the right side of the results page. These allow the user to filter the search results, showing only entries for people, or for organizations, etc.

Solr within VIVO

VIVO is based around an RDF triple-store, which holds all of its data. However, there are some tasks that a search engine can do much more quickly than a triple-store. Some of the fields in the Solr search index were put there specifically to help with these tasks.

For example, the browse area on the home page shows how many individuals VIVO holds for each class group.



VIVO could produce this data by issuing a SPARQL query against its data model. However, this would take several seconds for a large site, and we do not want the user to wait that long to see the home page. To avoid this delay, the class group of each individual is stored in the Solr record for that individual. Solr can count these fields very quickly, so VIVO issues a Solr query against the index, and displays the results on the home page.

Record counts on VIVO's index pages are obtained using the same type of Solr query.

[Home](#)
[People](#)
[Organizations](#)
[Research](#)
[Events](#)

People

[Cornell Faculty Member](#)
(2,774)

[Non-Faculty Academic](#) (4,778)

[Librarian](#) (155)


[Non-Academic](#) (4,496)

[Faculty Member Emeritus](#)
(693)

Cornell Faculty Member

[All](#)
[A](#)
[B](#)
[C](#)
[D](#)
[E](#)
[F](#)
[G](#)
[H](#)
[I](#)
[J](#)
[K](#)
[L](#)
[M](#)

page [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#) [31](#) [32](#) [33](#) [34](#) [35](#) [36](#)



[Abawi, George Samuel](#)
 Professor, Plant Pathology at G

[Abdul Razak, Intan Shameha Binti](#)

How is Solr created and configured?

The VIVO distribution includes a copy of Solr WAR file. When VIVO is installed the Solr WAR file is deployed to Tomcat as a web application.

The behavior of Solr depends extensively on its configuration files. These are stored in a directory that is called the Solr Home directory. In the standard VIVO installation, this is the `solr` sub-directory in the VIVO Home directory. When VIVO is installed, the Solr configuration files are copied to the Solr Home directory. When VIVO runs, the Solr search index is built inside the Solr Home directory.

If you are installing VIVO in a servlet container other than Tomcat, or if you are installing Solr in a separate servlet container, you will need to tell Solr how to find its home directory. See the instructions in [Building a VIVO distribution for other servlet containers](#).

The search index

What is in the index?

The Solr search index contains one record for each Individual in VIVO, unless that individual is explicitly excluded from the index. Exclusions are usually made for individuals that represent "context nodes" in the VIVO data model.

For example, if a professor teaches a course, the search index will contain:

- a record for the professor
- a record for the course

The VIVO data model also contains an individual that represents this teaching activity. That individual will be excluded from the index, since users would almost certainly prefer to find the teacher or the course in their search results, rather than the concept that connects the two.

What is in each record?

Each record in the search index contains several fields (see the chart below). The most commonly used field is `alltext`. In the record for a faculty member, `alltext` will contain her name, the name of her department, the names of her classes, the names of her papers and grants, etc. So, if you search for "Carpenter", you might see results for people named Carpenter, people in the Carpentry department, people who have written papers about carpentry, or have worked on grants about carpentry. You would also see results for the department itself, for the papers, and for the grants.

Solr index fields, VIVO 1.6

DocId	nameRaw	PREFERRED_TITLE
URI	nameText	siteURL
ALLTEXT	nameLowercase	siteName
ALLTEXTUNSTEMMED	nameLowercaseSingleValued	THUMBNAIL
classgroup	nameUnstemmed	THUMBNAIL_URL
type	nameStemmed	indexedTime
mostSpecificTypeURIs	acNameUntokenized	timestamp
BETA	acNameStemmed	etag
PROHIBITED_FROM_TEXT_RESULTS	NAME_PHONETIC	

In progress

- When an individual is added/edited/deleted, Solr is given the new information and updates the index.
- Sometimes the index must be rebuilt
 - Most commonly, after an ingest, since some of the ingest mechanisms bypass the usual VIVO framework
 - It would be too slow to update the Solr index on each new statement from the ingest
 - Working to add a search-aware ingest method, which Harvester or other tools could use.
 - A rebuild is done on the side, then replaces the previous index, and Solr switches to the rebuilt one.
 - send it requests to
 - search
 - add, update, or delete records

Customizing the index

In progress

- Building the record
- Exclusions

How does VIVO contact Solr?

In progress

- Need to tell VIVO how to contact Solr
 - Authorization tests, now obsolete
- VIVO may start before Solr does. Usually does.