AllegroGraph Web View — User Manual

The AllegroGraph Web View is an interface for exploring, querying, and managing AllegroGraph triple stores through a web browser.

Installation

Simply download the appropriate file from http://agraph.franz.com/agwebview/, and unpack it somewhere.

Running Web View

To use Web View, a server program has to be started, either on your own computer or on a server somewhere. This server is then accessed through a web browser. The `agwebview` (`.exe` on windows) file is used to start such a web-server.

Most of the configuration can be done through the browser, but some options have to be provided when starting the server. These are passed as command-line options to `agwebview`.

`--port` number
: The port on which to run the HTTP server. Defaults to 8080.

`--account` name:password
: Causes the server to create an administrative account using the given name and password. You'll have to pass this the first time you start the server, or you won't have an account to configure it.

`--publish` name:directory[:read-only]
: Publishes the triple store in the given directory under the given name. When ":read-only" is prepended the store can not be modified through the Web View interface. This option can be passed multiple times. Stores can also be published through the web interface.

`--cache` directory
: Determines where persistent settings and user accounts should be saved. Defaults to 'site-cache' in the installation directory.

`--log` file name
: Specifies where logging output should be written.

`--debug`
: Run the server in debug mode.

`--no-local-stores`
: Disable opening of local stores on this server.

`--no-remote-stores`

Disable opening of remote stores.

`--no-new-stores`

Do not allow users to create new triple stores.

One could run, for example, this:

```
agwebview --port 5000 --account me:xyzzy
```

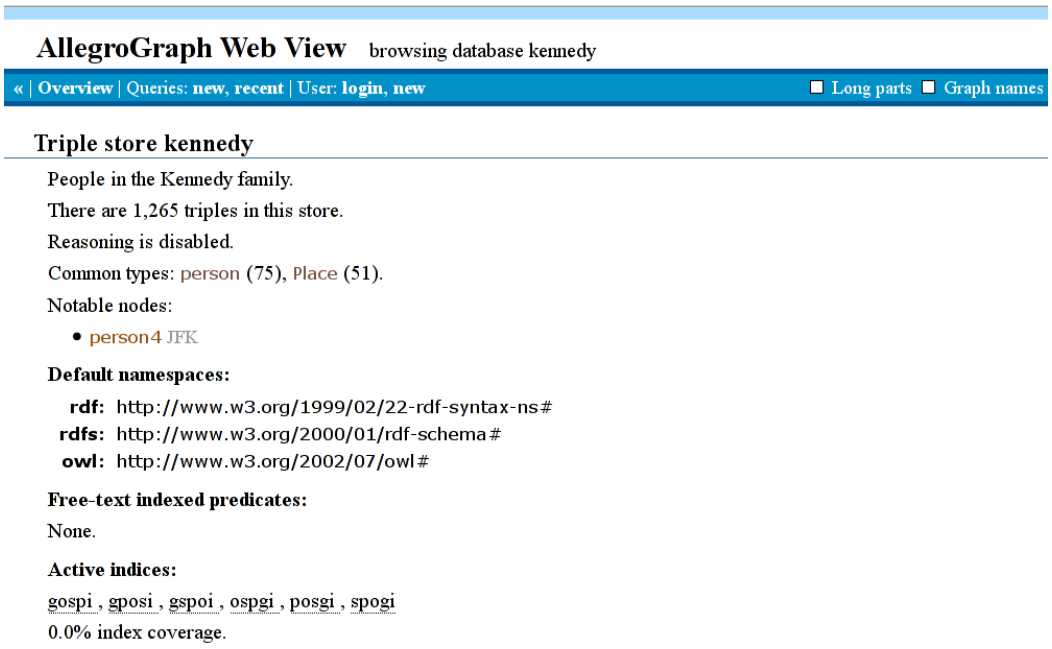
Now, going to http://localhost:5000 presents you with the front-page of the Web View. There is not much to be seen here yet, since we haven't opened any stores. To start configuring the site, click 'login' in the bar at the top of the screen, and use the administrative account we created: user name 'me', password 'xyzzy'.

After logging in, a lot more controls are visible. Stores can be added to the site, and settings can be modified. Note that all the configuration options and user accounts are stored on disk, in the `site-cache` directory in the installation directory, and will be remembered when the server is restarted. Use the `--cache` command-line option to run a server that saves its configuration in a different place.

If you have an AllegroGraph store somewhere on your system, try opening it. Enter its path in the 'File' field of the 'Open a local store' form, and give it a name. If you have no store, create an empty one instead. After clicking 'Create', a link to the store should appear on the top of the page.

User interface

When opening a store in Web View, you start at the store's overview page. This lists some information about the store, and, when logged in with an administrator account, provides the interface for managing the store and the Web View settings for that store.



A store overview

A store has a description, a set of 'notable' nodes (good starting points for users), a set of pre-defined queries, and some default namespaces. These can be used to make it easier for others to get familiar with the store, but are not much use if you are just using Web View to explore a store yourself.

Throughout the interface, a SPARQL-style notation is used for RDF resources and literals. `<http://example.com>` is used for URLs, which an be abbreviated to something like `rdf:type` (note the absence of < and >) if a suitable namespace has been defined. Literals are `"quoted"`, optionally with a type or language postfixed like `"10"^^xsd:integer` or `"ten"@en`. When the 'long parts' toggle (visible in the navigation bar) is enabled, resources are always displayed in full. When it is disabled, a shortened form is displayed (mousing over them gives a tooltip with the full form). Resources always link to their 'context' page, which shows a list of triples that the resource is involved in.

Relations with person4 » as the subject.

Predicate	Object
has-parent	person1
has-parent	person2
rdf:type	person
first-name	"John"
middle-initial	"Fitzgerald"
last-name	"Kennedy"
suffix	"none"
alma-mater	Princeton
alma-mater	Harvard
birth-year	"1917"
death-year	"1963"
sex	male
spouse	person5
has-child	person20
has-child	person22
has-child	person24
profession	navy
profession	congressman
profession	senator
profession	president
birth-place	place5

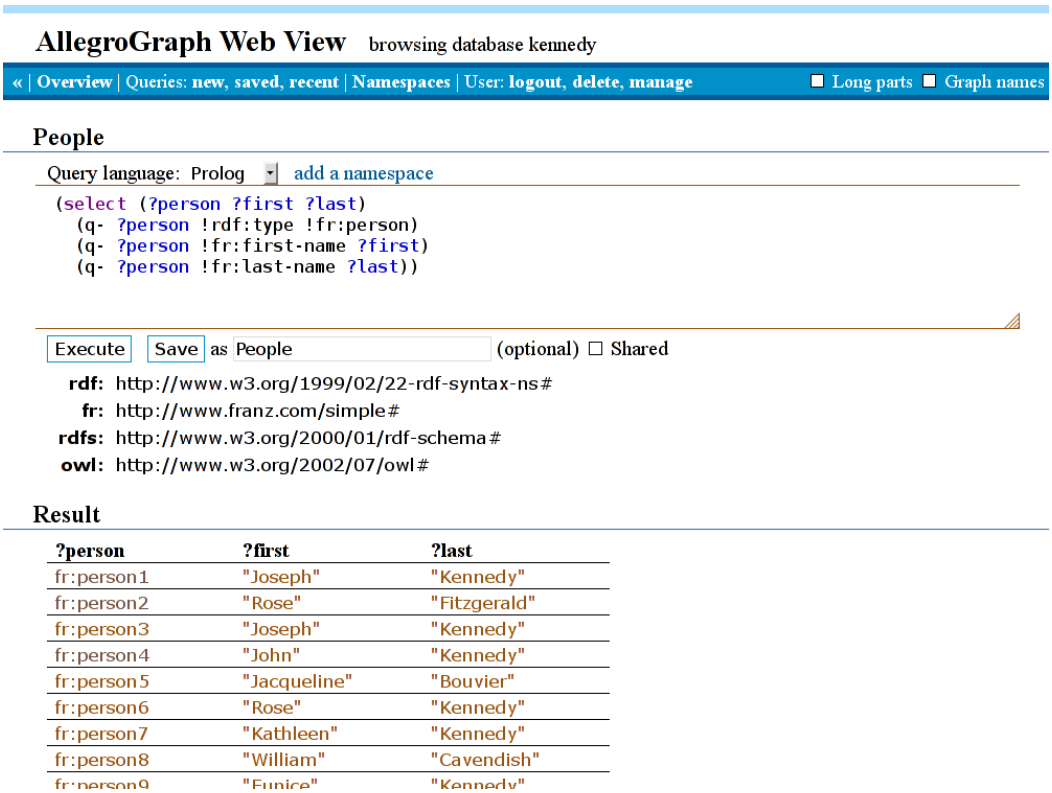
Relations with person4 » as the object.

Subject	Predicate
person1	has-child
person2	has-child
person20	has-parent

Node context view

At any time, a set of namespaces is active, which are used to shorten URLs. When not logged in, only the default namespaces are active. Logged-in users can configure the set of namespaces they want to use by clicking the 'namespaces' link in the top bar.

Queries can be executed by clicking 'Queries: new' in the top bar. This brings up an empty input area where a SPARQL query can be entered. When Prolog queries are enabled for the site, there is a menu that can be used to set the query language to Prolog instead. An overview of recently executed queries is available under 'Queries: recent'.



Executing a Prolog query

At any time, a query can be given a name and saved. This causes the query to appear in the 'Queries: saved' list, and to get a fixed URL. If the 'shared' check-box was checked when saving the query, this URL can also be accessed by others. Administrative users can add their saved queries to the store (from the 'Queries: saved' list), to make them appear in the store overview.

Graph names are not displayed by default, but when they are toggled on (see the check-box in the top bar), views that show triples (node context and text search results) will also show the named graph that each triple belongs to.

The search box in the top right of the screen (only visible if the triple store uses free-text indexing) will search the predicates that are indexed, displaying a list of matching triples. Note that it does not search *all* literals.

When a Google key is set for the site, data using AllegroGraph's geospatial format can be mapped with Google Maps:

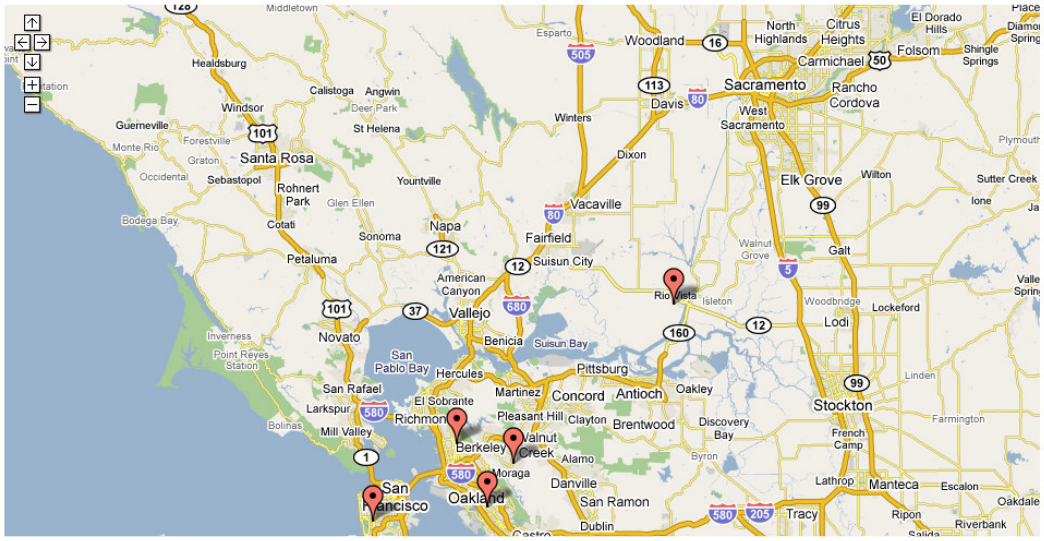
show namespaces, add a namespace

```
select ?name ?coords {
  ?place fr:coords ?coords ;
         fr:name ?name ;
         fr:state "California".
}
```

Execute Save as California (optional) ☐ Shared

Result

« Back to table view.



Mapping query results

Edit query

show namespaces, add a namespace

```
construct {
  ?c fr:has-parent ?p
} where {
  ?c fr:has-parent ?p
}
```

Execute Save as (optional) ☐ Shared

Result

« Back to table view.

Drag nodes to rearrange them.Showing 50 of 78 triples. Show more: 10, 25. (Layout out lots of triples can get slow.)

fr:person7

Visualizing the results of a CONSTRUCT query