Using OpenSearch interface in CatalogConnector

CatalogConnector now implements the OpenSearch interface[1], and its 'Geo' [2] extension. This note includes two main sections, the first one expains how to use it (from a user's point of view), and the second one explains how is it implemented (for developers and application administrators).

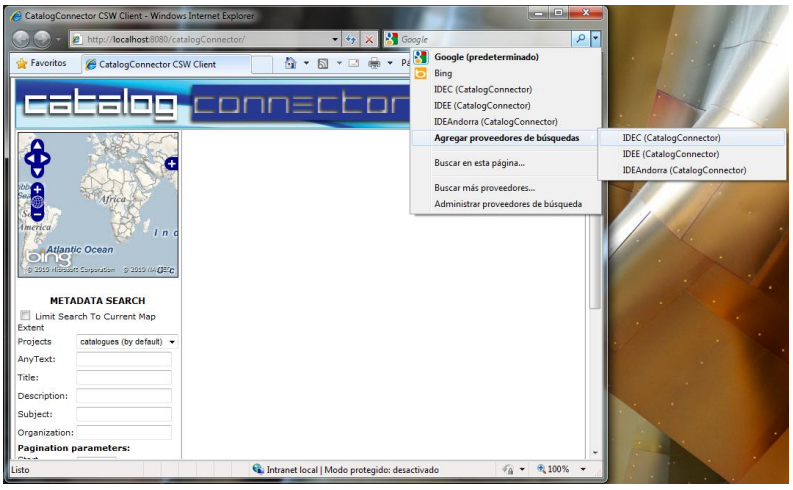
Using the OpenSearch interface

This first section shows how to use OpenSearch as a user. No knowledge on OpenSearch standard or CatalogConnector internals is needed to follow this section.

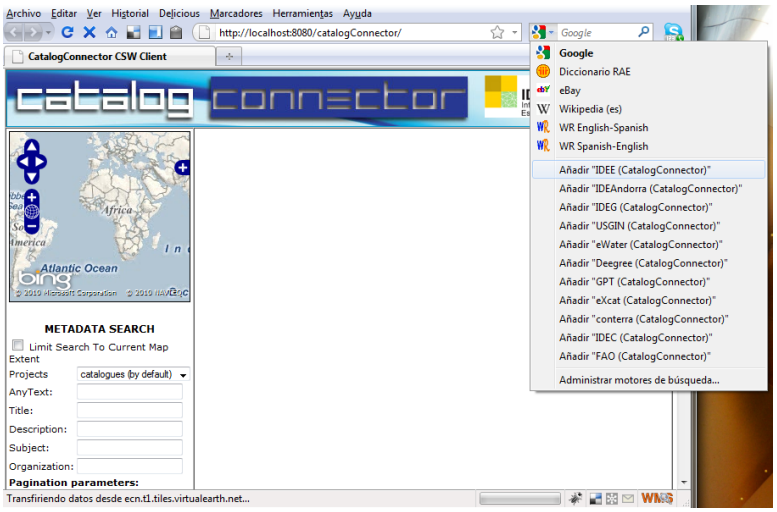
Catalog search in web browsers

OpenSearch interface can be used to personalize the 'custom search bar' present in modern web browsers. This search bar acts as a shortcut to search for specialized, thematic content, such as definitions in dictionaries, articles in stores, news, or help systems.

To add CatalogConnector's catalogs to a web browser, just load the application's main page, and click over the custom search bar icon to see the search bar menu:

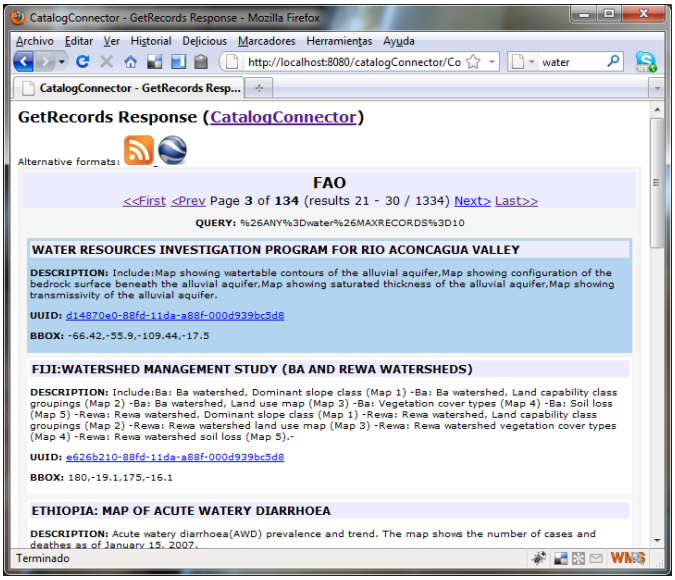


1 Adding catalogs in Internet Explorer search bar



2 Adding catalogs in Firefox search bar

The results for a query through the browser's custom search bar is an HTML document:

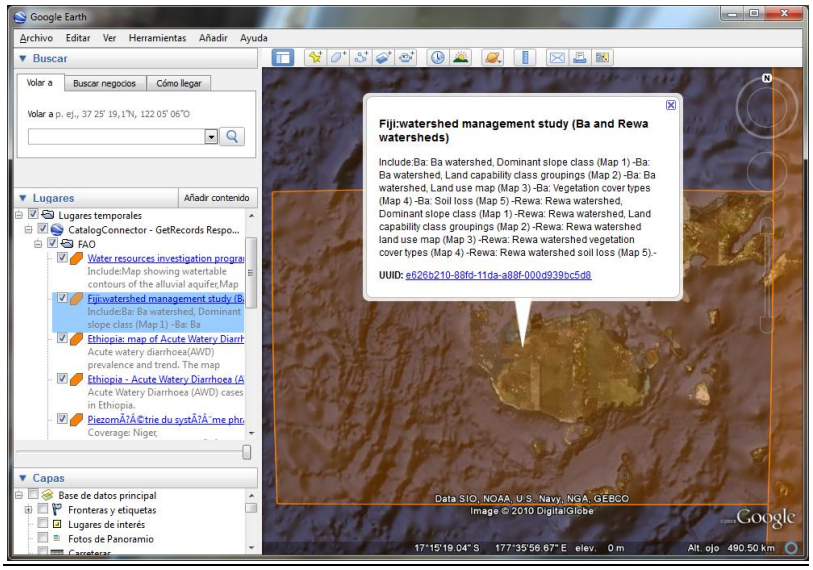


3 Results page

The results page contains links to alternative representation formats: Atom (for syndication) and KML (for 2D and 3D map viewers). Results are paged, so the "first", "previous", "next" and "last" links can be used to browse across result pages. Each item contains a title, a description, its unique identifier and the bounding box, if any. Clicking on the unique identifier link, the full (original) metadata record in XML format can be retrieved.

Result visualization in Google Earth

Clicking on the KML shows the actual results page in Google Earth.

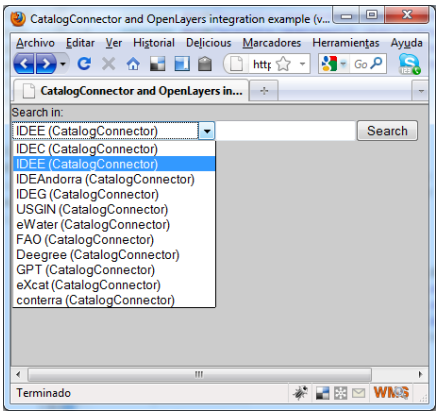


4 KML Results in Google Earth

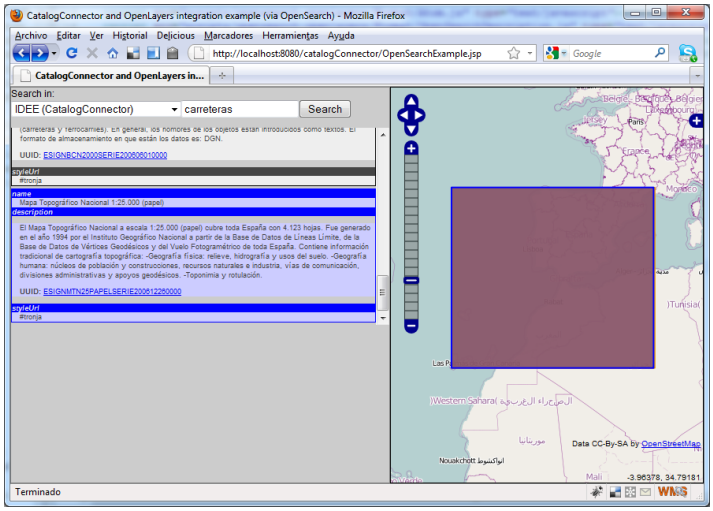
As in HTML, the UUID links to the full metadata record.

Search integration in OpenLayers-based viewers

An OpenSearch control for OpenLayers is under development [3]. CatalogConnector includes an example on how to use it.



5 Catalog selection in OpenLayers Search Control



6 Results view in OpenLayers Search Control

To include the control, some javascript files have to be included after the OpenLayers main library:

```
<script src="scripts/opensearch-openlayers/Format/Atom.js" type="text/javascript"></script>
<script src="scripts/opensearch-openlayers/Format/OpenSearchDescription.js" type="text/javascript"></script>
<script src="scripts/opensearch-openlayers/Strategy/OpenSearch.js" type="text/javascript"></script>
<script src="scripts/opensearch-openlayers/Control/OpenSearch.js" type="text/javascript"></script>
```

Then, an array with a collection of OpenSearch description document URLs must be created. CatalogConnector generates this array automatically with a new request command called “getOpenSearchDescription”:

http://localhost:8080/catalogConnector/Connector?Request=GetOpenSearchDescription

Finally, the OpenSearch control has to be instantiated and added to the map control:

```
map.addControl(new OpenLayers.Control.OpenSearch({
  div : $('os'),
  descriptions: descriptionsArray }));
```

Implementation details

This section is intended for CatalogConnector developers willing to understand how OpenSearch has been implemented. Knowledge on CatalogConnector source and the OpenSearch specification [1][2] is required.

Description documents

CatalogConnector’s main servlet includes a new request type: **GetOpenSearchDescription**.

- When called without any other parameter, it returns a JSON array with a collection of URLs. Each one points to the description document for each of the catalogs configured in the application instance.
  http://localhost:8080/catalogConnector/Connector?Request=GetOpenSearchDescription
- This collection of pointers can be retrieved also in HTML format. It is not a full HTML page, but the collection of <link> elements to be included in the HTML head as autodiscovery tags.

http://localhost:8080/catalogConnector/Connector?Request=GetOpenSearchDescription&Format=HTML

- In combination with the “Catalogues” parameter, it generates the OpenSearch description document itself, for the catalog or collection of catalogs stated in the “Catalogues” parameter.
  http://localhost:8080/catalogConnector/Connector?Request=GetOpenSearchDescription&Catalogues=IDEC

The OpenSearch description document contains the template URLs used by OpenSearch clients to build the search queries, along with other metadata: Author, contact info, keywords, title, icons, etc. This common information is stored in:

```
WebContent/WEB-INF/OpenSearchDescriptionTemplate.xml
```

Request parameters and response formats

Results can be obtained in the following formats: JSON, XML, Atom (with GeoRSS), KML and HTML. The two former (JSON and XML) are the original CatalogConnector response formats. Atom, KML and HTML have been added for OpenSearch functionality, but can be used in a “GetRecords” request, just as the original ones:

http://localhost:8080/catalogConnector/Connector?Request=GetRecords&Project=catalogues&outputFormat=HTML&catalogues=IDEE,FAO&any=rios

These three new formats are generated internally through a XSLT transformation of the XML response. XSLT transformations are stored in:

```
WebContent/scripts/getRecords2ATOM.xsl
WebContent/scripts/getRecords2HTML.xsl
WebContent/scripts/getRecords2KML.xsl
```

Some additional information needed for pagination and linking but not present in the XML getRecords response is passed directly from the Java code. These parameters are:

- getRecordByIdBaseURL: Needed for CSW full record link generation.
- startIndex/itemsPerPage: Needed to build “prev”, “next”, “first” and “last” pagination links.
- SEARCH: A link to the self OpenSearch description document, useful for autodiscovery.
- SELF: A link to the results page itself.
- ATOM/HTML/KML: Links to alternative format representations of results page.

References

[1] OpenSearch 1.1 (draft 4): http://www.opensearch.org/Specifications/OpenSearch/1.1

[2] OpenSearch Geo extensión (draft 1): http://www.opensearch.org/Specifications/OpenSearch/Extensions/Geo/1.0/Draft_1

[3] OpenLayers OpenSearch control: http://geoportal.dlsi.uji.es/OpenSearch/