IEDA Integrated Catalog Geoportal deployment and configuration

[**Configuration**](#_467kdw6y6pae) **2**

[Connecting geoportal with Elasticsearch](#_9xo848rv07ks) 2

[Login configuration, API permissions](#_6gjkr8982hf2) 3

[Geoportal Users](#_adlftm8i0v8m) 4

[Metadata schema configuration](#_2v4lybpwt9jr) 4

[Language localization](#_7v3tmc6cdj67) 5

[User interface customizations:](#_7wtnn99rnv5s) 5

[Configuring facets](#_suthe6i1o8gf) 6

[Configuring header, footer](#_unm7ru7bhz14) 7

[Content for about page](#_f7z6uney2qn4) 7

[Configuration of result cards](#_nm1iux3ekenn) 7

[**Elasticsearch operations**](#_b4qgcoy5vlbu) **7**

[**How to delete records in catalog**](#_st3hgc9do8km) **8**

[**Development environment:**](#_fyf1csuiy4rt) **8**

[**Deployment:**](#_bxd10frlgtef) **9**

[Server information:](#_q5pkeatp82w9) 9

[Redirecting 80 to 8080](#_8yf6e76qo1hp) 10

[Restarting Geoportal](#_ef4xro22osct) 10

[Restarting ElasticSearch](#_bgvib0gv0zof) 10

[**Harvest tool**](#_r4zcpl4mo3re) **11**

[**Synchronization with partner catalogs**](#_6sdlgrdp9l9b) **11**

[ECL sync](#_ejjdjypb601j) 12

[USAP-DC sync](#_b1m0pebzakkn) 12

The IEDA integrated catalog is implemented using the (ESRI) Geoportal v2.5 software. This software was originally developed by ESRI, starting around 2008, and was made an open-source project, with a first github release in 2013. The IEDA implementation is a configuration forked from the v2.5 master at <https://github.com/Esri/geoportal-server-catalog>. The IEDA fork is in the IEDA organization github as a public repository at <https://github.com/iedadata/geoportal-server-catalog>.

The master branch is the IEDA integrated catalog, and SouthernOceanAntarctic branch is the spcatalog.

These should be cloned to your local machine.

The Geoportal application is designed to harvest metadata in a wide variety of metadata interchange formats, index the content using Elasticsearch, and provide a search interface with free text search, a map widget for specifying geographic bounding box search, and a collection of facets assisting users with browse search approach and results filtering.

Dependencies:

Geoportal v2.5 from iedadata/geoportal-server-catalog

Harvester. Use 2.5.1 release from <https://github.com/Esri/geoportal-server-harvester/releases/tag/v2.5.1>

Elasticsearch v5.6.6

Tomcat (v 8)

The production and dev IEDA servers can are deployed via Amazon EC2 virtual machines.

URLs:

app.iedadata.org/catalog and catalog.iedatadata.org redirect to

catalog.iedadata.org/geoportal.

The Antarctic and Southern Ocean Geoportal is on the same production server at catalog.iedadata.org/spcatalog.

# Configuration

## Connecting geoportal with Elasticsearch

Connection to an accessible ElasticSearch instance is documented at <https://github.com/Esri/geoportal-server-catalog/wiki/Elasticsearch-configuration>.

ElasticSearch configuration is at /etc/elasticsearch/elasticsearch.yml

Settings:

*cluster.name: iedageoportal*

*node.name: iedaelk*

*path.data: /var/lib/elasticsearch*

*path.logs: /var/lib/elasticsearch*

*network.host: localhost*

*http.port: 9200*

These are scattered amongst copious comments in the yml file.

The geoportal configuration document is at geoportal/src/main/resources/config/app-context.xml (in the gitHub repository). The IEDA geoportal ElasticSearch is configured with the following parameters:

Elastic context:

*<beans:property name="clusterName" value="iedageoportal" />*

*<beans:property name="indexName" value="metadata" />*

*<beans:property name="indexNameIsAlias" value="false" />*

*<beans:property name="autoCreateIndex" value="true" />*

*<beans:property name="allowFileId" value="false" />*

*<beans:property name="mappingsFile" value="config/elastic-mappings\_hierarchy.json" />*

*<beans:property name="nodes">*

*<beans:list>*

*<beans:value>LOCALHOST</beans:value>*

*</beans:list>*

*</beans:property>*

Harvester context defines host name where harvester is running:

<*beans:property name="nodes">*

*<!-- The list of host names within the Harvester cluster, one value element per host -->*

*<beans:list>*

*<beans:value>LOCALHOST</beans:value>*

*</beans:list>*

*</beans:property>*

The mappings file defines various field-name extensions that get mapped to default ElasticSearch index field types. ‘Elastic-mappings\_hierarchy.json’ is a copy of the mapping file from CINERGI that includes definition of some hierarchical fields, but none of these are used in the IEDA geoportal. The default mappings file could probably be used, but hasn’t been tested.

## Login configuration, API permissions

The configuration file geoportal/src/main/resources/config/app-security.xml configures various access control elements. Line 13 “ *<beans:import resource="authentication-simple.xml"/>*” sets up user authentication by simple password. See <https://github.com/Esri/geoportal-server-catalog/wiki/Security-configuration-Simple> for more information. There are other authentication template configurations to use LDAP, ArcGIS, or Oauth, but these have not been test for IEDA.

App-security.xml defines the roles that can access Get, Put, Post, Delete http requests on the API resources. The API is documented at <https://github.com/Esri/geoportal-server-catalog/blob/master/geoportal/doc/api.txt>

### Geoportal Users

User names, passwords and roles are defined in geoportal/src/main/resources/config/authentication-simple.xml lines 12-16:

*<security:user name="loginnamehere" password="passwordhere" authorities="ROLE\_ADMIN,ROLE\_PUBLISHER" />*

*<security:user name="namehere" password"passwordhere" authorities="ROLE\_PUBLISHER" />*

*<security:user name="user" password="pass" authorities="ROLE\_USER" />*

NOTE-- don’t forget to change the password and user name for the harvester output broker connecting to the Geoportal if you update the user name and password in the authentication-simple.xml file! The harvester has to connect either as ROLE\_ADMIN or ROLE\_PUBLISHER; current configuration is connecting as ROLE\_ADMIN.

Need to create this file in your local github clone, using the authentication-simple-template.xml file.

## Metadata schema configuration

A configuration file is required for each input metadata format and usage profile. The configuration files define how each metadata format is recognized (evaluator.js), and passes the document to a component that parses the format (e.g. evaluatorISO.js). The base evaluator file (geoportal/src/main/resources/metadata/js/Evaluator.js) has been updated to IEDA process gmd (ISO19115/ISO19139) and gmi (ISO19115-2/ISO19139-2) metadata with the same parser.

// IEDA update: make interrogation path //gmd:MD\_Metadata instead of /gmd:MD\_Metadata; that will capture gmd:MD\_Metadata elements that are in other wrappers

// change the details XSLT to the ISO19139ToHTML.xsl presentation, an updated presentation from USGIN

// Skip schema validation so can process invalid documents

// toKnownXslt is null, could insert path to XSLT for pre processing here; initial testing to try and harvest DataCite XML by transforming to ISO 19139 XML failed, this doesn’t seem to work. See <https://github.com/Esri/geoportal-server-catalog/issues/121>.

*"iso19115": {*

*key: "iso19115",*

*evaluator: G.evaluators.iso,*

*interrogationXPath: "//gmd:MD\_Metadata",*

*identifier: "http://www.isotc211.org/2005/gmd",*

*detailsXslt: "metadata/details/iso-details/ISO19139ToHTML.xsl",*

*//xsdLocation: "http://schemas.opengis.net/csw/2.0.2/profiles/apiso/1.0.0/apiso.xsd",*

*toKnownXslt: null*

},

// updates same as iso19115 evaluator

*"iso19115-2": {*

*key: "iso19115-2",*

*evaluator: G.evaluators.iso,*

*interrogationXPath: "//gmi:MI\_Metadata",*

*identifier: "http://www.isotc211.org/2005/gmi",*

*detailsXslt: "metadata/details/iso-details/ISO19139ToHTML.xsl",*

*//xsdLocation: "http://ngdc.noaa.gov/metadata/published/xsd/schema.xsd",*

*schematronXslt: null*

*}*

The G.evaluators.iso parser (evaluator) is defined in the file EvaluatorFor\_ISO.js (same directory as Evalutor.js). This program maps content from xpaths in the ISO xml document to content in the ElasticSearch index fields. New index fields can be defined, but if the fields are changed, the index will need to be purged and rebuilt (see [instructions below](#_b4qgcoy5vlbu)).

Customizations in [EvaluatorFor\_ISO.js](https://github.com/iedadata/geoportal-server-catalog/blob/master/geoportal/src/main/resources/metadata/js/EvaluatorFor_ISO.js) are documented with comments in the javascript.

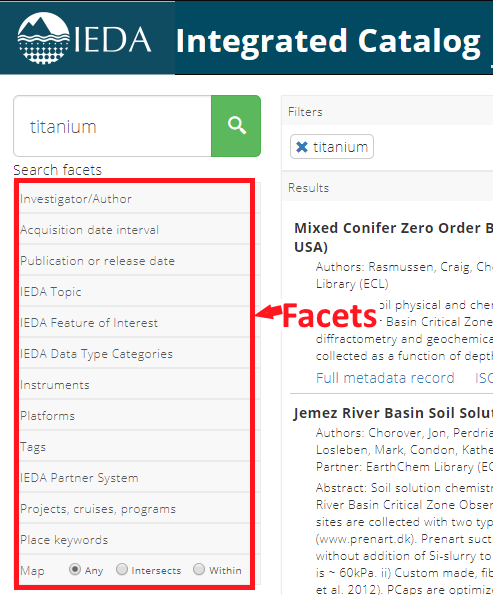
## Language localization

The geoportal is implemented with a resource file containing a javascript object that maps UI text items to the strings that should be presented. This allows language localization by substituting resource files that have strings in any language. The resource file is located at geoportal/src/main/webapp/app/nls/resource.js. References to the language-localized string resources look like this: *${i18n.nav.signIn}* in the html templates. In some of the IEDA customizations, these references to the resource.js file have been replaced by strings for convenience, since we don’t plan on releases in different languages.

## User interface customizations:

The web site is constructed using the Dojo framework. In the source code directory, the web site is defined in the webapp/app directory. The webapp/app/main directory contains definitions of the major web pages. The webapp/images directory contains the logos displayed on the web pages. In the app/main directory, there is a %PageName%.js file for each page, and a templates directory containing a %PageName%.html file for each page.

### Configuring facets



The search facets displayed on the left side of the search page are configured in geoportal/src/main/webapp/app/main/templates/SearchPanel.html. This page defines a set of html div elements, one for each facet. The data-dojo-type for the div references javascript components in the app/search directory that implemented the behavior for each facet. Each div has a set of data-dojo-props (properties) that bind the facet to a field in the Elasticsearch index (‘field’, ‘toField’), the label for the facet (‘label’), and optional tooltip for the facet. Some facet types get other properties as well. In particular, a ‘props’ attribute containing a javascript object specifying the ‘size’ (number of elements to show in the facet list) and sort order (‘order’). Default sorting is by frequency of occurrence of given facet value.

### Configuring header, footer

The application main page frame, including a header and footer and the top level functionality tabs (Search and About) are defined in geoportal/src/main/webapp/app/main/templates/App.html. Comments in that file indicate customizations made, which include the main application title, inserting the IEDA logo, the color of the header and footer bar, hiding the link to the ‘Map’ panel, and moving the admin sign in into the ‘hamburger’ drop down menu in upper right.

The footer element is inserted separately in the Search and About html template pages. During development the footer was initially placed on the App page, but didn’t work with the way the Map page was implemented, so the footer was placed on the pages separately. This is not currently necessary since the Map page has been disabled, but the footer hasn’t been moved back to the App page.

### Content for about page

The contents of the about page are currently rendered in an html file at geoportal/src/main/webapp/app/main/templates/AboutPanel.html. This text was by hand crafted in html from an out of the box example page.

### Configuration of result cards

The presentation of search results is via a ‘Card’ object. The text displayed in the card heading was modified to show authors, creation date and the source repository with updates to code in lines 480-535 of geoportal/src/main/webapp/app/search/ItemCard.js. Font size on the card was increased by editing the geoportal/src/main/webapp/app/style/main.css file, lines 249 and 266.

# Elasticsearch operations

When the geoportal starts, it checks to see if the configured index is present in the linked Elasticsearch cluster. If an index with the IndexName specified in geoportal/src/main/resources/config/app-context.xml is not present, a new index is created based on the fields defined in the evaluator javascript modules in the geoportal/src/main/resources/metadata/js directory. If new index fields are defined in an evaluator function, the Elasticsearch index must be purged and Tomcat restarted to build a new index. All metadata will need to be reharvested once Tomcat is started.

To purge the index, make an HTTP Delete request to the elasticsearch index at

[http://username:password@%catalog host location%/geoportal/elastic/metadata.](about:blank)

Where ‘metadata’ is the name of the index. The host location currently is catalog.iedadata.org:8080..

# How to delete records in catalog

* login with an Admin account (see iedaadmin information above). A new drop-down menu labeled 'Options' will appear on each item card, to the right of 'Links to resource'.
* Click on the 'Options' for the item you wish to remove, and select 'Delete record'. The record will be removed from the catalog.

# Development environment:

Modifications to the Geoportal application were done in the source code in a clone of the iedadata/geoportal-server-catalog repository, using Eclipse JEE Oxygen 4.7.2. To create the project:

start eclipse

(use a new workspace folder, didn't test with other existing workspace)

file-->import--import

select dialog opens

pick Maven--Existing Maven Projects

Maven Projects dialog opens

'from directory' at top, click 'Browse...' and find geoportal-catalog-server gitHub repo and open

should see /geoportal/pom.xml..

and /geoportal-search/pom.xml... as check box options.

check /geoportal/pom.xml....

imports maven project, then validates, takes a few minutes

when it’s done, right click on geoportal, select export-->WAR file is an option

The WAR should be 46 Mb.

Commits were made frequently to document the changes being made. Development was mostly done on a separate branch. Pull requests were used to sync with some updates from the ESRI master.

# Deployment:

## Server information:

Discussion here is for deployment on Amazon VM

Tomcat manager login:

URL: <http://xxxxxxx>

username xxxxxxx xxxxxx manager-gui admin-gui

### Redirecting 80 to 8080

port redirect so geoportal works on 80 (i.e. catalog.iedadata.org)

When you did the commands you follow them up with --permanent so that the changes are persistent on reboot.

sudo firewall-cmd --zone=public --add-masquerade  
sudo firewall-cmd --zone=public --add-masquerade --permanent  
  
sudo firewall-cmd --zone="public" --add-forward-port=port=80:proto=tcp:toport=8080  
sudo firewall-cmd --zone="public" --add-forward-port=port=80:proto=tcp:toport=8080 --permanent

## Restarting Geoportal

Restart Geoportal by restarting the the host Tomcat instance. Log in to the server.

$ sudo systemctl restart tomcat

If the war file in the webapps directory has been updated since last restart, it will be unpacked and deployed as a new Geoportal build.

## Restarting ElasticSearch

Same as geoportal:

$ sudo systemctl restart elasticsearch

Check that its running:

$curl -X GET '<http://localhost:9200>'

Should get

*{*

*"name" : "iedaelk",*

*"cluster\_name" : "iedageoportal",*

*"cluster\_uuid" : "CwQk80-7S3S75LZBhExJdQ",*

*"version" : {*

*"number" : "5.6.6",*

*"build\_hash" : "7d99d36",*

*"build\_date" : "2018-01-09T23:55:47.880Z",*

*"build\_snapshot" : false,*

*"lucene\_version" : "6.6.1"*

*},*

*"tagline" : "You Know, for Search"*

*}*

# Harvest tool

Metadata harvesting into the geoportal is managed by a separate application named ‘Harvester’. The IEDA Integrated Catalog is currently using release 2.5.1, obtainable at <https://github.com/Esri/geoportal-server-harvester/releases>. To install, unzip the archive there, copy the harvester.war file to the Tomcat webapps directory, and restart Tomcat.

The harvester operates by executing tasks that collect metadata from an input broker and pass it to an output broker for processing. Input brokers provide connection to various kinds of service end points. The IEDA Integrated Catalog content is currently harvested using the Web Accessible Folder (WAF) broker. Output brokers connect to several kinds of end points: Arcgis portal (not used by IEDA), Geoportal v2.5 (IEDA Geoportal), and a folder. Harvesting into a local folder is useful for testing and debugging the harvest process.

Current endpoints:

Input brokers

ECL from IEDA: Web accessible folder. <http://get.iedadata.org/metadata/iso/ecl/>

MGDL from IEDA: Web accessible folder. <http://get.iedadata.org/metadata/iso/mgdl/>

USAP from IEDA: Web accessible folder. http://get.iedadata.org/metadata/iso/usap/

Output Broker

Localhost: This is the IEDA Integrated Catalog Geoportal, running on the same Tomcat instance as the harvester. <http://localhost:8080/geoportal/>. User name iedaadmin, pwd : *!#DA@dm1n?*

Harvests can be scheduled by clicking on the ‘schedule’ link in the ‘tasks’ view of the harvester.

<http://catalog.iedadata.org/harvester/>.

To see a record of harvest results, click on the ‘tasks’ tab (left side menu) in the harvester, and click on the ‘history’ link to the right of the harvest job you are interested in.

The harvest process updates all records that are in the harvest source. If a previously harvested record is removed from the harvest source, it remains in the geoportal catalog. Records that are to be removed must be manually deleted from the geoportal. See ‘[how to delete records](#_st3hgc9do8km)’.

# Synchronization with partner catalogs

ISO XML records get harvested to the IEDA Integrated Catalog Geoportal by the harvester (see [Harvest Tool](#_r4zcpl4mo3re), above), from web accessible folders on the seafloor server, accessed via get.iedadata.org URL. The ISO xml files should be deposited at

for USAP: /public/mgg/web/get.iedadata.org/htdocs/metadata/iso/usap.

for ECL: /public/mgg/web/get.iedadata.org/htdocs/metadata/iso/ecl

for MGDS: /public/mgg/web/get.iedadata.org/htdocs/metadata/iso/mgdl

The file-name construction is not critical, the gmd:fileIdentifier property in the xml is used to correlate harvested records with existing records. File names should have ‘.xml’ at the end.

USAP-DC, ECL, and MGDL all maintain their own metadata databases. MGDL currently pushes ISO records to the metadata/iso/mgdl directory on get.iedadata.org, and the sync to the IEDA Integrated Catalog (IIC) is taken care of by weekly harvests from the metadata/iso/mgdl directory.

The original plan was to get a configuration for the Geoportal to harvest DataCite XML directly for ECL and USAP-DC, but there is no existing configuration for that format in the evaluator portfolio in the Geoportal Github, and the ‘toKnownXslt’ function in the harvest evaluator isn’t working ([see GitHub issue](https://github.com/Esri/geoportal-server-catalog/issues/121)), so for now plan B is to transform the DataCite XML records using the [DataCite to ISO19139 xml transformation](https://github.com/iedadata/resources/blob/master/DataCiteXMLTransforms/DataciteToISO19139v3.2.xslt) on hand to transform the records and write them to the appropriate metadata/iso watch directory on Github. The transform has some logic to identify USAP, ASP@LDEO, or ECL records, see the section defining the ‘datacentre’ variable, lines 44-68 in the xslt code. This might need to be updated if the encoding of the DataCite XML is changed, or if records from some other IEDA partner are to be processed. For now, if a partner system is not identified the source system is listed as IEDA.

Different workflows are used to sync ECL and USAP-DC metadata for the integrated catalog.

## ECL sync

DataCite records are written to get.iedadata.org/metadata/doi as individual records named with the 6 digit number that is the final token in the DOI, e.g. ‘100066’. This was not always true prior to early 2018. Lulin has extended the script that writes new or updated DataCite records in the /metadata/doi folder to transform them using the xslt [DataciteToISO19139v3.2.xslt](https://github.com/iedadata/resources/blob/master/DataCiteXMLTransforms/DataciteToISO19139v3.2.xslt) to generate an ISO19139 XML record and this is then written to the metadata/iso/ecl directory to keep the metadata in sync. The integrated catalog harvests daily from /metadata/iso/ecl.

## USAP-DC sync

As part of the [USAP-DC curator workflow](http://www.usap-dc.org/curator/help), the curator will generate ISO XML using the DataCite XML extracted from the USAP-DC database and the [DataciteToISO19139v3.2.xslt](https://github.com/iedadata/resources/blob/master/DataCiteXMLTransforms/DataciteToISO19139v3.2.xslt) file. This can then be viewed and edited, if needed. The curator will save the file in the watch directory /web/usap-dc/htdocs/watch/isoxml on the USAP-DC server. A cron job on the USAP-DC server will check this directory daily and rsync it with the /public/mgg/web/get.iedadata.org/htdocs/metadata/iso/usap directory on seafloor, where it will be harvested by the Geoportal.

Code can be found in the USAP-DC repo:

<https://github.com/iedadata/usap-dc-website/blob/master/usap.py>

<https://github.com/iedadata/usap-dc-website/blob/master/lib/curatorFunctions.py>

<https://github.com/iedadata/usap-dc-website/blob/master/bin/makeISOXMLFile.py>

<https://github.com/iedadata/usap-dc-website/blob/master/inc/isoxml_cron>