

# OpenWIS Documentation



*Manual for*

*Met Office Government UK*



Author: GeoCat bv

Last Updated: 9/11/2015

# Table of Contents

## [1. Developer's manual](#)

### [1.1. Building GeoNetwork](#)

#### [1.1.1. Set up local environment](#)

##### [1.1.1.1. Prerequisites](#)

##### [1.1.1.2. Clone Git Repository](#)

##### [1.1.1.3. Clone Git Submodules](#)

#### [1.1.2. Generate war](#)

### [1.2. Configure and Deploy](#)

#### [1.2.1. Configure using the Source Code](#)

##### [1.2.1.1. Configuration for OpenWIS](#)

#### [1.2.2. Configure using a provided war](#)

#### [1.2.3. Deployment](#)

#### [1.2.4. \[Optional\] Upgrade with Upstream changes](#)

### [1.3. Extending Customizations](#)

#### [1.3.1. Translations](#)

#### [1.3.2. Graphics with D3JS](#)

#### [1.3.3. Source Code for OpenWIS extensions](#)

##### [1.3.3.1. Direct Download](#)

## [2. Administration manual](#)

### [2.1. ADMIN1 Blacklisting](#)

### [2.2. ADMIN2 Browse Product Metadata Table](#)

### [2.3. ADMIN3 Browse OpenWIS cache content](#)

### [2.4. ADMIN4 Monitoring Statistics](#)

### [2.5. ADMIN5 Configura Dissemination Channels for Groups](#)

### [2.6. ADMIN5 Browse Subscriptions](#)

### [2.7. DM1: Default metadata insertion](#)

[2.8. AAC: SAML2 authentication](#)

[2.9. MM1: Create custom metadata identifiers](#)

[2.9.1. Metadata identifier templates](#)

[2.10. MM2 Configure OAI-PHM limit](#)

[2.11. MM3 Metadata Categories and Category based harvesting](#)

[3. User's manual](#)

[3.1. Create metadata](#)

[3.2. UI1 Subscription Management](#)

[3.3. UI2 My Downloads Page](#)

[3.4. DSS](#)

[3.4.1. New Subscription](#)

[3.4.2. Ad-hoc request for delivery](#)

[3.4.3. Direct Download via HTTP](#)

# 1. Developer's manual

## 1.1. Building GeoNetwork

### 1.1.1. Set up local environment

#### 1.1.1.1. Prerequisites

To set up the local environment you have to have installed on your system the software **Git**, **Maven** version 3+<sup>1</sup> and a **Java JDK** version 7+.

#### 1.1.1.2. Clone Git Repository

The first thing to do is localize the repository url where your customized version of GeoNetwork is stored. For example, if you are going to use the version stored on GeoCat's repository, the url will be ***git@eos.geocat.net:openwis/openwis-4.git***

If we are using the command line, we just have to place the terminal on the folder where we are going to explode the GeoNetwork source code and run:

```
cd $path  
git clone git@eos.geocat.net:openwis/openwis-4.git
```

This command will download from the repository not only the plain source code but also the git configuration and part of the history of the repository. The output of the terminal should look like this:

```
$ git clone git@eos.geocat.net:openwis/openwis-4.git geonetwork  
Cloning into 'geonetwork'...  
remote: Counting objects: 166664, done.  
remote: Compressing objects: 100% (57140/57140), done.  
remote: Total 166664 (delta 91319), reused 165825 (delta 90728)  
Receiving objects: 100% (166664/166664), 155.49 MiB | 4.07 MiB/s, done.  
Resolving deltas: 100% (91319/91319), done.  
Checking out files: 100% (9493/9493), done.
```

#### 1.1.1.3. Clone Git Submodules

GeoNetwork contains several submodules that are usually not cloned by default. To make sure we have the latest version available of this submodules on our local source code, we have to run the “submodule init” and a “submodule update” git commands.

If we are using the command line, we just have to place the terminal on the root folder of the GeoNetwork source code and run:

```
cd $path/geonetwork;  
git submodule init;
```

---

<sup>1</sup> Official Documentation available at <https://maven.apache.org/install.html>

### ***git submodule update***

After a similar output than on the previous step, we will have all the source code available on our local machine, ready to be built:

```
$ git submodule init; git submodule update
Submodule 'e2e-tests/chromedriver' (https://github.com/geonetwork/chromedriver.git)
registered for path 'e2e-tests/chromedriver'
Submodule 'geoserver' (https://github.com/geonetwork/core-geoserver.git) registered for
path 'geoserver'
Submodule 'web-ui/src/main/resources/catalog/lib/style/bootstrap'
(https://github.com/twbs/bootstrap.git) registered for path
'web-ui/src/main/resources/catalog/lib/style/bootstrap'
Submodule 'web-ui/src/main/resources/catalog/lib/style/font-awesome'
(https://github.com/FortAwesome/Font-Awesome.git) registered for path
'web-ui/src/main/resources/catalog/lib/style/font-awesome'
Cloning into 'e2e-tests/chromedriver'...
remote: Counting objects: 21, done.
remote: Total 21 (delta 0), reused 0 (delta 0), pack-reused 21
Unpacking objects: 100% (21/21), done.
Submodule path 'e2e-tests/chromedriver': checked out
'501f7852e8620c4f33de37cf0911f73bf2d2d002'
Cloning into 'geoserver'...
remote: Counting objects: 460, done.
remote: Total 460 (delta 0), reused 0 (delta 0), pack-reused 460
Receiving objects: 100% (460/460), 62.82 MiB | 9.90 MiB/s, done.
Resolving deltas: 100% (169/169), done.
Submodule path 'geoserver': checked out
'1eb4c68418e44ac36da5cb425124118799f75e49'
Cloning into 'web-ui/src/main/resources/catalog/lib/style/bootstrap'...
remote: Counting objects: 73124, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 73124 (delta 1), reused 0 (delta 0), pack-reused 73120
Receiving objects: 100% (73124/73124), 85.84 MiB | 10.51 MiB/s, done.
Resolving deltas: 100% (44536/44536), done.
Submodule path 'web-ui/src/main/resources/catalog/lib/style/bootstrap': checked out
'a365d8689c3f3cee7f1acf86b61270ecca8e106d'
Cloning into 'web-ui/src/main/resources/catalog/lib/style/font-awesome'...
remote: Counting objects: 32710, done.
remote: Compressing objects: 100% (410/410), done.
remote: Total 32710 (delta 325), reused 0 (delta 0), pack-reused 32300
Receiving objects: 100% (32710/32710), 18.32 MiB | 6.01 MiB/s, done.
Resolving deltas: 100% (17591/17591), done.
Submodule path 'web-ui/src/main/resources/catalog/lib/style/font-awesome': checked out
'2649d91d18629bab071449b3bc4cb02761037a57'
```

### 1.1.2. Generate war

Once you have the source code on your local machine, you can invoke maven to run. If we are using the command line, we just have to place the terminal on the root folder of the GeoNetwork source code and run:

```
export MAVEN_OPTS="-Xmx512M -XX:MaxPermSize=256M"  
mvn clean package install -DskipTests
```

Be patient, because this command may take long to finish, as it has to download from maven artifacts repository many third party libraries. When the build finishes, you should see something like this:

```
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 04:43 min  
[INFO] Finished at: 2015-07-28T13:16:36+01:00  
[INFO] Final Memory: 194M/407M  
[INFO] -----
```

The generated war file we are interested in is located on **web/target/geonetwork.war**.

[Optional] You can try the build by moving to the web folder and run a Jetty server from there:

```
cd web; mvn jetty:run-exploded -DskipTests; cd ..
```

You can test your deployment by opening a browser and entering <http://localhost:8080/geonetwork>

## 1.2. Configure and Deploy

### 1.2.1. Configure using the Source Code

There are many things that can be configured in GeoNetwork<sup>2</sup>, but the basic thing to configure on a deployment is the database GeoNetwork is going to use.

First check which type of database is used. Choose the database type to use on the file **web/src/main/webapps/WEB-INF/config-node/srv.xml**. Then update the **web/src/main/webapps/WEB-INF/config-db/jdbc.properties** file with connection information.

If we have done changes on the configuration files, we have to rebuild the war file. This is done by invoking again the **mvn package install -DskipTests** command. Make sure you run it on the root folder of GeoNetwork's source code.

---

<sup>2</sup> See <http://geonetwork-opensource.org/manuals/trunk/eng/users/administrator-guide/index.html> for more information.

#### 1.2.1.1. Configuration for OpenWIS

On the special case of the OpenWIS installation, we also have to modify an extra file to point to the OpenWIS SOAP service. This configuration is on

***openwis-services-client/src/main/resources/config-spring-geonetwork.xml***

There you have to replace all the endpoint urls pointing to <http://localhost:8061/> with the URL where the SOAP service is deployed.

#### 1.2.2. Configure using a provided war

There are many things that can be configured in GeoNetwork<sup>3</sup>, but the basic thing to configure on a deployment is the database GeoNetwork is going to use. To configure the database, you have to modify a couple of files contained inside the war file.

First check which type of database is used. Choose the database type to use on the file ***WEB-INF/config-node/srv.xml***. Then update the ***WEB-INF/config-db/jdbc.properties*** file with connection information.

Once you have modified this files, make sure you place them again inside the ***geonetwork.war*** file.

#### 1.2.3. Deployment

To deploy your GeoNetwork instance, copy the ***geonetwork.war*** file to your application server. In the case of Tomcat, there should be a folder called ***webapps*** which contains all the applications to deploy. This is the folder where the ***geonetwork.war*** file should be copied in.

Once the application server is running, you have to make sure the metadata profiles you are going to use are enabled on your installation. To do this, you have to be logged in as an administrator user. The default administrator user is admin/admin.

After login, you have to enter the administration interface by clicking on the “***Admin Console***” on the toolbar. Then you select the “***Metadata and Templates***” option, which will display the list of schemas already available to be enabled. On that list, you can select the schemas you want to enable on your installation. You can select more than one schema at the same time.

Once you have selected all the schemas you plan to use, you can click on both buttons at the bottom of the list. The first one will load the templates, which will be used when you create a new metadata based on those schemas. The second button will load the samples for the schemas, which will be added to the catalog:

---

<sup>3</sup> See <http://geonetwork-opensource.org/manuals/trunk/eng/users/administrator-guide/index.html> for more information.

Load samples and templates for metadata standards

## Standards available

1 selected▼

### Dublin Core - CSW

Metadata records produced by CSW services.

### Dublin Core

The Dublin Core metadata standard

### Geographic information - Methodology for feature cataloguing (ISO 19110:2005)

ISO 19110 standard for describing Feature Types

### Geographic information - Metadata (ISO/TS 19139:2007)

ISO19139 metadata standard

Load templates for selected standards

Load samples for selected standards


Now that you have your schema profiles enabled on the server, you can create your own metadata based on this schemas.

To create a new metadata, click on the “**Contribute**” link of the top toolbar. This will lead you to the editor dashboard, where you will be able to see all the current metadata records and templates. If you click on the “**Add new record**” button, you will be redirected to a new view where you can select the type of metadata you want to create. First, select the type of metadata and then select the template that fits better with your use case.



## Create a

Create a  
**Others**

  
Others

From **Vector - Layer Template**

Generic Layer Template

**Vector - Layer Template**

WMS - Layer Template

WMTS - Layer Template

+ Create

▼

When you click on the “**Create**” button, you will be redirected to the metadata editor which will display your newly created metadata based on the template you have chosen.

### 1.2.4. [Optional] Upgrade with Upstream changes

You will probably want to keep your source code upgraded with the latest upstream changes. The easiest way to achieve this is to add a new remote to git with the official GeoNetwork repository and do frequent pulls from it.

So, we add a new remote to our local git repository:

```
git remote add upstream https://github.com/geonetwork/core-geonetwork.git
```

Now, check the version of GeoNetwork on which your source code is based in. This is defined on the pom.xml files of your source code. At the moment of writing this documentation, the source code is based on version 3.0.1, which means that we will want to keep upgraded with the 3.0.x branch of GeoNetwork repository.

So, to pull the changes from upstream and merge them to our source code, we run the following command:

```
git pull upstream 3.0.x
```

if there are no conflicts, we will be directly redirected to the commit message we want to use to mark this merge. If there are conflicts, we will have to solve them first before being able to mark the pull as merged.

Once the conflicts (if any) are solved and the commit is done, we can push to our repository the changes, so the merge is not only stored on our local machine:

```
git push
```

If you are using GeoCat's repository as the main repository (see *Clone Git Repository* at the beginning of this document), check that you have privileges to write to the repository. If not, the push will fail.

Don't forget to test the software before re-deploying it. Even when git doesn't detect conflicts, that doesn't mean that the functionality is not broken.

## 1.3. Extending Customizations

We have kept the source code of GeoNetwork with the minimum changes required to add new independent maven projects that contains all the customizations for openWIS. This projects are at the same level as the rest of the GeoNetwork maven projects, inside the root folder of the repository. All of them start with the prefix "openwis-".

The files on the project openwis-web-ui will override the files on the project web-ui, which is the default project of GeoNetwork. This way, we can modify parts of the user interface that will otherwise require modifying the web-ui project itself.

### 1.3.1. Translations

Localized strings are translated on JSON files inside the **web-ui/src/main/resources/catalog/locales** folder. Each file on this folder has the same name format: {lang}-{section}.json. The language is the two letter ISO code for languages. The section refers to where on the user interface is this file going to be used.

Each localized string has an id which is the one used on the user interface. You can add a new localized string on the user interface with the angular directive `data-translate=""`. For example, the following code will search for the "openwis-cache" id on the translation files and substitute that identifier with the associated string:

```
<div class="panel-heading" data-gn-slide-toggle="" data-translate="">openwis-cache</div>
```

General GeoNetwork's translations are managed on the Transifex platform<sup>4</sup>. If you find that some language you want to use doesn't have a complete translation, you can contribute to GeoNetwork by translating and improving translations on that platform. If there is enough interest<sup>5</sup> on new languages, we can also add a complete new language to this translation tool.

All the changes done on Transifex are pushed to the repository daily. So if you keep upgrading your repository with the official repository of GeoNetwork, the translations done on Transifex will be merged with the translations on your local code.

### 1.3.2. Graphics with D3JS

GeoNetwork works with the D3JS<sup>6</sup> library to generate graphs over statistical data. It is an html5 utility to generate all kind of graphs based on an array of values. You can see an example of how GeoNetwork works with D3JS on the **web-ui/src/main/resources/catalog/js/admin/DashboardSearchStatController.js** file.

---

<sup>4</sup> Internationalization of GeoNetwork in general: <https://www.transifex.com/geonetwork/core-geonetwork/>

<sup>5</sup> We only add new languages when some contributor shows motivation to, at least, translate a significant percentage of that language.

<sup>6</sup> <https://github.com/mbostock/d3/wiki/Gallery>

### 1.3.3. Source Code for OpenWIS extensions

To connect to the SOAP service of OpenWIS, maven generates automatically a set of SOAP client utilities on the project `openwis-services-client`. Additionally we have created a Java class for each endpoint inside the `org.openwis.*.client` packages. This utility classes are the link between GeoNetwork and the SOAP clients.

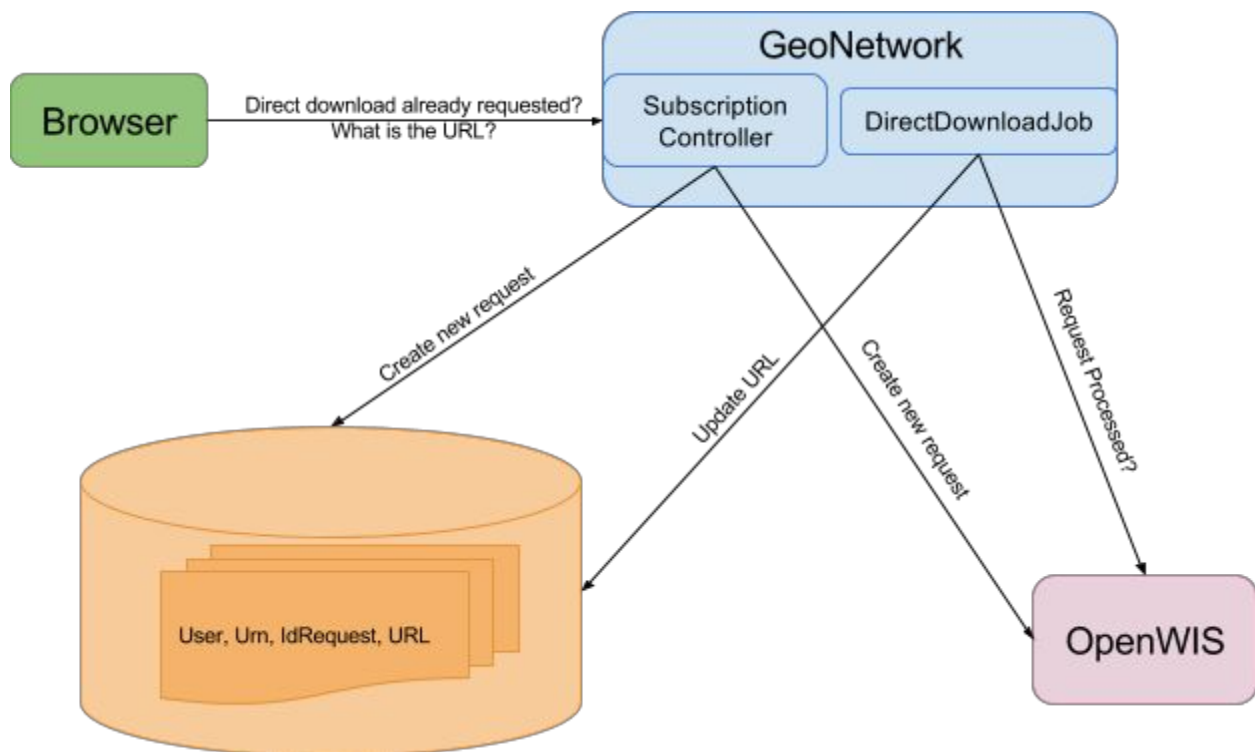
If the wsdl services definition change, a new build will be required with the right wsdl files updated on the folder `openwis-services-client/src/main/resources/schemas`. To configure the URL of the endpoints, open the `openwis-services-client/src/main/resources/config-spring-geonetwork.xml` file and check for the “defaultUri” properties. Each endpoint is configured independently.

The user interface works with backend controllers that converts the SOAP requests and responses to something more appropriate for the browser. This controllers can be found on the `openwis-services` project inside the `org.fao.geonet.services.openwis` package. All of them are Spring MVC classes that returns JSON data.

#### 1.3.3.1. Direct Download

To prevent lots of calls to the OpenWIS service, GeoNetwork will cache the direct download requests so we can have a fast and light response for each direct download button that has to be shown on the browser. This also prevents the user to do more than one request for the same urn.

There is a job called **DirectDownloadJob** that updates the values on the local database, so when OpenWIS has the url available, it will be quickly updated on the database so the SubscriptionController controller can tell the browser that the request has been processed and which is the URL to use.



### 1.3.3.2. Blacklisting Pagination

The user interface for browse and edit blacklisted users is based on DataTables<sup>7</sup>. To add pagination, we need to modify the Java class **org.fao.geonet.services.openwis.blacklisting.Service** available on openwis-services project. There, on the function **search** around line 81 we have a couple of lines commented we should fill with the required data:

```
// response.setRecordsTotal(client.getTotal());

// response.setRecordsFiltered(client.getTotalCurrentQuery(startWith));
```

This response object will return everything DataTables need for pagination.

The object client (**org.openwis.blacklist.client.BlackListClient**) is the facilitator to connect to the SOAP service. It uses some automatically generated classes during the build. This classes mimic the SOAP service on a Java way. If the WSDL changes (for example to add the above commented functionality), we have to update the schema file so maven can generate the new client classes on **openwis-services-client/src/main/resources/schemas/BlacklistService.wsdl**.

Once this automatically generated classes are available, we can complete the BlackListClient class with the appropriate calls and uncomment the lines in the Service class to enable pagination on the user interface.

### 1.3.3.3. My Downloads Page

The user interface for browse and edit downloads is also based on DataTables. To complete it, we need to create three new functions in the OpenWIS service to search, return the total number of records and return the total number of records filtered (if there is such filter). Once we have the new wsdl endpoint available, we have to update the schema files availables on

**openwis-services-client/src/main/resources/schemas/**

If the endpoint is new and there is no current wsdl available there, you will have to tell maven to process it to generate the associated classes. This is done by adding this new wsdl file on the **openwis-services-client/pom.xml** file the same way as the other files are mentioned there. Then, we have to configure this new endpoint on the **config-spring-geonetwork.xml** file inside **openwis-services-client project**. To achieve this, we have to enter three new beans on the file:

```
<bean class="org.openwis.$package.client.$PackageClient"><property name="webServiceTemplate"
ref="$packageWebServiceTemplate" /></bean>

<oxm:jaxb2-marshaller id="$packageMarshaller" contextPath="org.openwis.$package.client" />

<bean class="org.springframework.ws.client.core.WebServiceTemplate" id="$packageWebServiceTemplate" >

    <constructor-arg ref="messageFactory" />

    <property name="marshaller" ref="$packagemarshaller" />

    <property name="unmarshaller" ref="$packagemarshaller" />

    <property name="defaultUri" value="$url" /></bean>
```

<sup>7</sup> DataTables: <http://datatables.net/>

That definition will inject a new client class called ***org.openwis.\$package.client.\$PackageClient*** with the automatically generated classes that mimic the SOAP service. We have to create this class with all the facilitator functionality that we will need on the controller (search with filters, total number of records and total number of records filtered).

Once we have the client for the wsdl schema definition, we need to modify the Java class ***org.fao.geonet.services.openwis.subscription.SubscriptionController*** available on openwis-services project. There, we will probably require to create a new function copying the ***search*** one, but instead of calling ***manager.retrieveSubscriptionsByUsers***, we will have to call the appropriate facilitator class that connects to the new endpoint (***\$PackageClient***). The response object will return everything DataTables need for creating the user interface. Remember to assign a different ***@RequestMapping*** value to the new function in SubscriptionController, always starting the name with “***openwis***”.

Now, we just have to configure the user interface to use this new request mapping as the source of data. To do this, we open the ***gnOpenwisAdminRequestController.js*** file available in ***openwis-web-ui*** project and change the DataTable configuration with our new mapping:

```
$scope.dtOptions = DTOptionsBuilder.newOptions().withOption(  
    "sAjaxSource",  
    $scope.url + 'openwis.subscription.search?myself=true')
```

Now the user interface will get the data from the new endpoints through the new service (request mapping) defined in SubscriptionController.java.

## 2. Administration manual

On this section we will visit the most relevant admin operations relevant for the OpenWIS domain. To complete this guide, the official GeoNetwork documentation<sup>8</sup> covers the rest of the administration sections on this software.





### 2.1. ADMIN1 Blacklisting

Users can be blacklisted by OpenWIS. This blacklisting does not prevent the user to authenticate on the portal, just prevents the user to download products and interact with the OpenWIS service normally. Some users are blacklisted automatically based on bandwidth and usage, but they can also be banned and unbanned manually by an administrator user. The full list of blacklisted users is available on the administration console, on the ***User and Groups*** section:

---

<sup>8</sup> GeoNetwork documentation available here:  
<http://geonetwork-opensource.org/manuals/trunk/eng/users/administrator-guide/index.html>

[Manage groups](#)
[Manage users](#)
[BlackList](#)
[My Subscriptions](#)
[My requests](#)

BlackList					
Show <b>10</b> entries					Search: <input type="text"/>
User	Nb Warn	Vol Warn	Nb Blacklist	Vol Blacklist	Status
banneduser	500	5000	1000	10000	BLACKLISTED_BY_ADMIN 
greedyuser	500	5000	1000	10000	BLACKLISTED_BY_VOLUME_OF_DISSEMINATIONS 
niceuser	1000	3000	2000	4000	NOT_BLACKLISTED 
saveduser	500	5000	1000	10000	NOT_BLACKLISTED_BY_ADMIN 
User	Nb Warn	Vol Warn	Nb Blacklist	Vol Blacklist	Status
Showing 0 to 0 of 0 entries (filtered from NaN total entries)					
First Previous <b>1</b> 2 3 4 5 ... Next Last					

The administrator can click on the eye icon at the right of each row, opening a new modal dialog that shows all the attributes associated to that user. This attributes can be edited and saved to the database:

Map
 + Contribute
 Admin console
 admin admin (Admin)

### Edit Blacklist

Username	greedyuser
Nb Dissemination Warning Threshold	500
Vol Dissemination Warning Threshold	5000
Nb Dissemination Blacklist Threshold	1000
Vol Dissemination Blacklist Threshold	10000









☒ Blacklist

Save

Cancel

## 2.2. ADMIN2 Browse Product Metadata Table

An administrator user can check the list of all products available on the platform on the admin console, on the **Metadata and Templates** section. This product table offers filters based on all the columns available:

productMetadata							
Show <b>25</b> entries		Search: <input type="text"/>					
Metadata Identifier	Metadata Title	Metadata Category	Originator	Process	GTS Category	FNC Patte	Actions
urn:x-wmo:md:int.wmo.wis::CSBD01WBSB	Test for subscriptions, delivery and download		RTH focal point	LOCAL	Unknown		   
ca3f634f-e487-4577-8dc8-39338af4a913			RTH focal point	LOCAL	Unknown		   
<div> <div>Metadata Identifier</div> <div>Metadata T</div> <div>Metadai</div> <div>Originat</div> <div>Process</div> <div>GTS Cat</div> <div>FNC</div> <div>Actions</div> </div>							
Showing 1 to 2 of 2 entries				First Previous <b>1</b> Next Last			

Each product row has five different actions available:

- View Metadata
- View product details
- Edit Metadata
- Duplicate Metadata
- Remove Metadata

Both view metadata and edit metadata works as a default GeoNetwork instance, opening the view and the metadata editor. Same happens with remove and duplicate metadata, which calls the default GeoNetwork operations for to duplicate and remove metadata respectively.

On the View product details, all the attributes associated to the product will be shown. Some of this attributes can be modified and saved accordingly on the OpenWIS SOAP service.



## Edit Product Metadata

**Metadata URN:**  
**Metadata Title:**  
**Originator:**

urn:x-wmo:md:int.wmo.wis::CSBD01WBSB  
  
 RTH focal point

---

**GTS Category:**Unknown  
  
**Data Policy:**unknown  
  
**FNC Pattern:**  
  
**GTS Priority:**  
  
**File Extension:**  
  
**Ingested:**  

Add Ingestion Filter

**Overriden GTS Category:**  
**Overriden Data Policy:**  
**Overriden FNC Pattern:**  
**Overriden GTS Priority:**  
**Overriden File Extension:**  
**Fed:**  

Add Feeding Filter

Save

Cancel

### 2.3. ADMIN3 Browse OpenWIS cache content

An administrator user can enter the **Statistics** tab on the administration console to search through the cached files on the OpenWIS service:



▼ Browse Cache

Show **10** entries Search:

File name	Urn	Checksum
A_IUPK14AMMC030826_C_AMMC_20151103082618_f0Psnckqrnhhyudkuh.txt	urn:x-wmo:md:int.wmo.wis::IUPK14AMMC	318757c51ec1e7b6837d3b9d1
A_IUPK14AMMC030826_C_AMMC_20151103082619_f10Ppiqtehxvpsclwdkn.txt	urn:x-wmo:md:int.wmo.wis::IUPK14AMMC	cbf505159968c68a8a2a6073c
A_IUPK14AMMC030826_C_AMMC_20151103082619_f11Phdqmmngpgwhawgej.txt	urn:x-wmo:md:int.wmo.wis::IUPK14AMMC	2fd6b4c995203c4e25e5a6ff11
A_IUPK14AMMC030826_C_AMMC_20151103082619_f12Pjhxniiijxgwbgc.txt	urn:x-wmo:md:int.wmo.wis::IUPK14AMMC	2f11ae11bc049d79cc58fd027
A_IUPK14AMMC030826_C_AMMC_20151103082619_f13Peqctvmfuggonbpx.txt	urn:x-wmo:md:int.wmo.wis::IUPK14AMMC	dc3056ce882d6e068b9b1971
A_IUPK14AMMC030826_C_AMMC_20151103082619_f14Pqbdnjhsvgnswfdpt.txt	urn:x-wmo:md:int.wmo.wis::IUPK14AMMC	d540c2a8bcd9c7eb2a1d1d70

## 2.4. ADMIN4 Monitoring Statistics

Next to the Browse Cache tab, we can find the OpenWIS statistics tab, which also shows a table with all the statistics available on the OpenWIS SOAP service:

Ingested Data Statistics

Ingested Data Statistics ▼  
Show **25** entries Search:

date	size
22/10/2015	473500
23/10/2015	94700
26/10/2015	94700
28/10/2015	94700
29/10/2015	94700
30/10/2015	94700
3/11/2015	189400

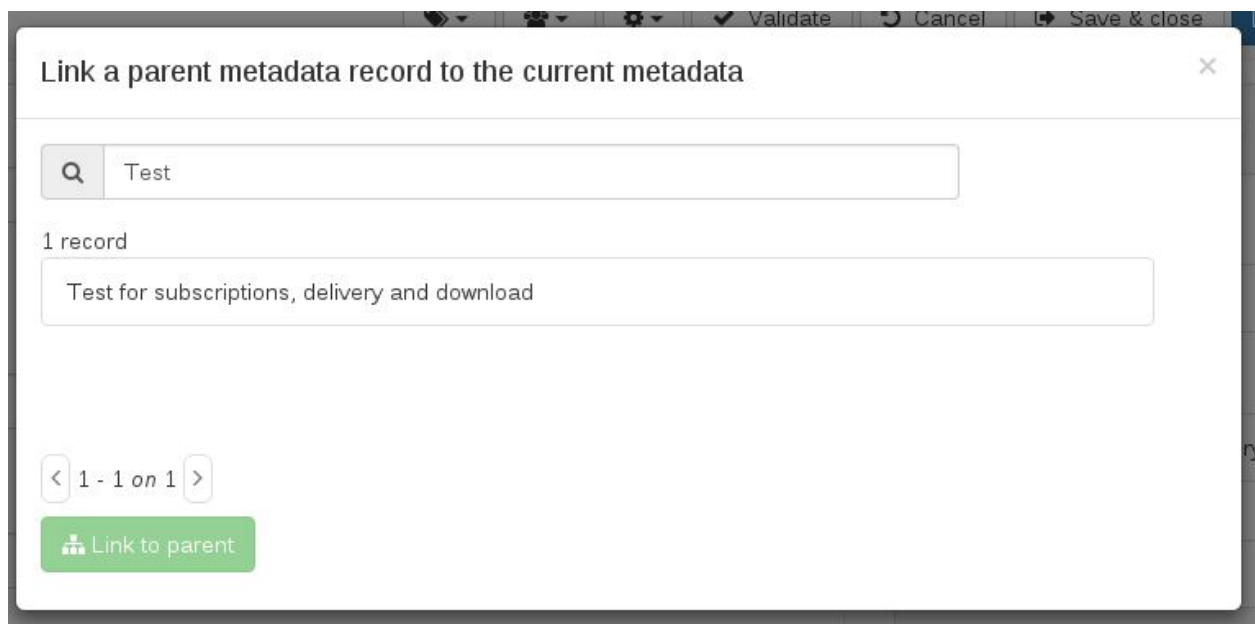
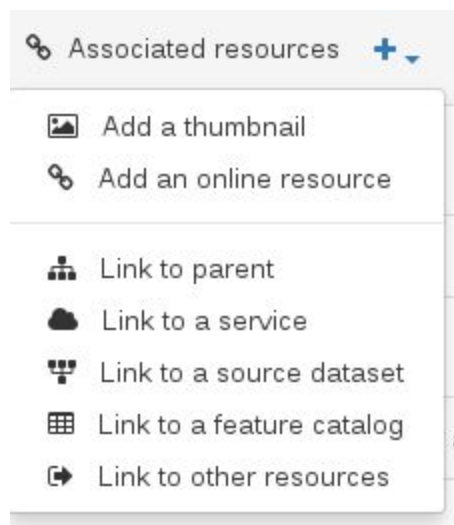
This table offers four types of statistics:

- Ingested Data Statistics
- Disseminated Data Statistics
- Exchanged Data Statistics
- Replicated Data Statistics

## 2.5. ADMIN5 Configure Dissemination Channels for Groups

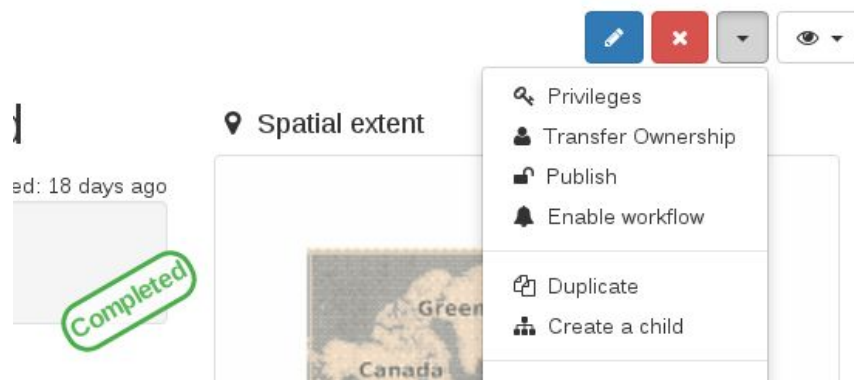
Sometimes a user will subscribe or request a deliver to several products at the same time. This is done through the parent/child relationship on the metadata which defines the dissemination channels. A dissemination channel is defined by a metadata on GeoNetwork who is the parent of all the products we want to add to the channel.

So, the first step to create a dissemination channel is to create the metadata parent. This is done through the same **“Contribute”** top menu as when creating metadata for products. Once we have created the metadata for the Channel, we edit one by one all the products we want to add to the Channel and we assign the Channel metadata as the parent metadata of the product with the **“Associated Resources”** option on the right menu.

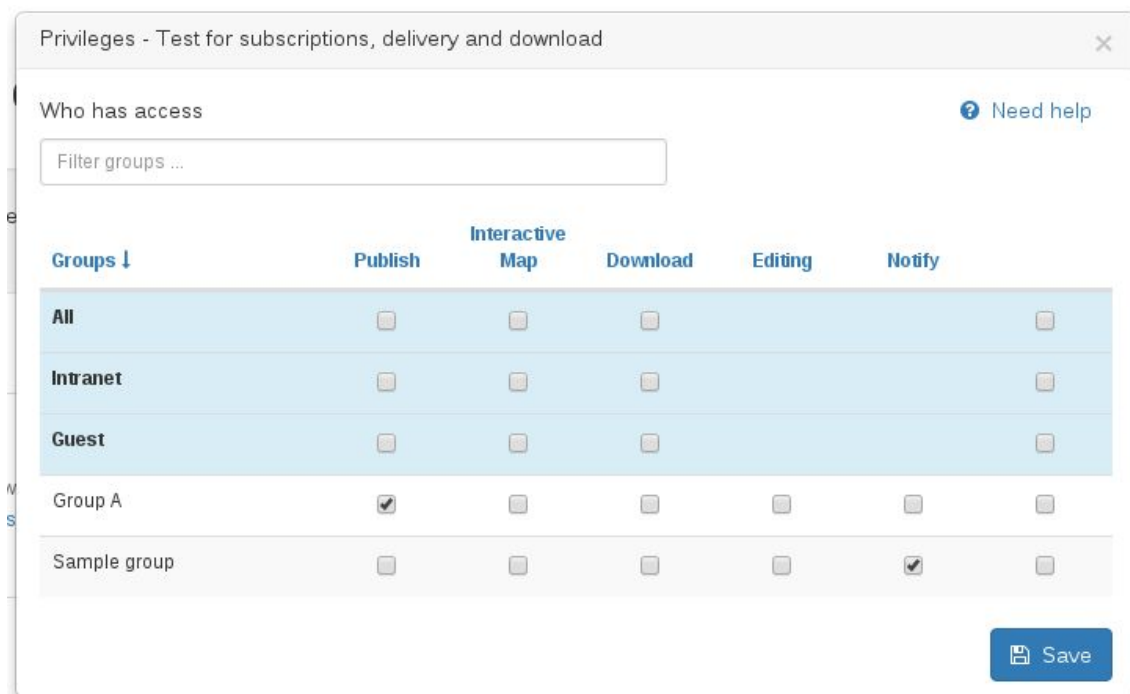


Now, when the user subscribes or request a deliver on the parent metadata channel dissemination, he will subscribe or request a deliver to all the children products associated to that metadata.

If we want to assign special privileges for the dissemination channel, to filter which groups can see which channels, we can do it by assigning GeoNetwork privileges over the dissemination channel metadata. This is done on the metadata actions menu (which appears on the metadata view, the editor dashboard and the search results):



On the “Privileges” option, the privileges modal window will be shown so the administrator can select which groups can see the metadata:



Also, if a user has privileges over only certain children products on a dissemination channel, he will be subscribed only to the products he has access to. The privileges on each product are handled the same way.

## 2.6. ADMIN5 Browse Subscriptions

An administrator user can browse all the subscriptions available on the OpenWIS service on the administration console, **Metadata and Templates** section. This table can be filtered by the groups available on GeoNetwork, so only the subscriptions associated to users on that group will be shown.

[Metadata & templates](#)
[Formatter](#)
[Schematron](#)
[productMetadata](#)
[Manage Subscriptions](#)

Manage Subscriptions

Group

Show  entries

Title	User	Urn	Status	Backup	Starting Date
Test Subscriptions	admin	b396a71f-3e22-40d5-ae4c-d18c000ef5b7	ACTIVE		
	admin	urn:x-wmo:md:int.wmo.wis::CSBD01WBSB	SUSPENDED		
	admin	urn:x-wmo:md:int.wmo.wis::CSBD01WBSB	ACTIVE		

Showing 1 to 3 of 3 entries

First Previous  Next Last

Each subscription row has four actions:

- View Product
- Edit Subscription
- Pause/Resume subscription
- Discard Subscription

## 2.7. DM1: Default metadata insertion

An authenticated user (with edit privileges) can create a new metadata using the following endpoint:  
<http://.../geonetwork/srv/eng/md.insert>

This service works with the following parameters:

- data=\$plain\_xml
- group=\$id\_of\_the\_group

The rest of the parameters are optional:

- template: y|n
- styleSheet: \$to\_apply\_to\_xml
- validate: on|off
- schema: \$schema\_of\_the\_metadata (example: iso19139)
- category: \$to\_assign (example: application)

## 2.8. AAC: SAML2 authentication

SAML can be activated with two steps:

1. Enable SAML authentication on the config-security.xml file inside WEB-INF/config-security folder by uncommenting the line that contains the `<import resource="config-security-shibboleth.xml"/>`
2. Configure the header name used for the username on the file config-security-shibboleth-overrides.properties inside the folder WEB-INF/config-security. This header name is the property `shibbolethConfiguration.usernameKey`

It is also recommended that other headers are added to this authentication. For example, if no email is provided, some features may not work right for this user.

## 2.9. MM1: Create custom metadata identifiers

This feature allows to customise the behaviour of GeoNetwork when creating new metadata:

- Create a random UUID. This is the "standard" behaviour of GeoNetwork and it's set as default.
- Use metadata URN templates to define the metadata identifiers. This option allows the user to enter a metadata identifier from the metadata URN templates defined by an administrator user.

To allow the users to use custom metadata identifiers, go to *Administration > Settings > Metadata create* and **disable** the option **Generate UUID**:

### Metadata create

Generate UUID



GeoNetwork creates assigns a random metadata UUID to the metadata created by a user (default). To assign the metadata UUID manually disable this option and fill the metadata UUID prefix.

### 2.9.1. Metadata identifier templates

This feature allows to manage the metadata identifier templates available in the metadata create page (see Create Metadata in the User's manual).

To maintain the metadata identifier templates go to *Administration > Metadata & templates*.

Settings
Logo
Sources
CSW
Virtual CSW
CSW test
Map servers
Metadata URN templates

Metadata URN templates

Filter

BoM DCPC Metadata

WMO Core profile URN

+ New metadata URN template

WMO Core profile URN
x Delete
Save

**Name**

WMO Core profile URN

**Template**

urn:x-wmo:md:int.wmo.wis::TT}{AA}{II}{CCCC}

For each template a user can define some parameters to fill in the metadata create form. Parameters are defined between curly braces. For example:

- urn:x-wmo:md:int.wmo.wis::TT}{AA}{II}{CCCC}
- au.gov.bom::IDCODE}

The parameters defined between curly braces are rendered in the metadata create page as form fields:

Metadata Uniform Resource Name

**Template**

WMO Core Profile URN
⬆

urn:x-wmo:md:int.wmo.wis:: TT AA  
II CCCC

**TT**

**AA**

**II**

**CCCC**

## 2.10. MM2 Configure OAI-PMH limit

This feature allows to configure the number of records returned (defaults to 10) but the OAI-PMH service requests.

To change this setting, go to the *Administration > Settings > Open Archive Initiative (OAI-PMH) Provider* and set the value for the field **Maximum records**.

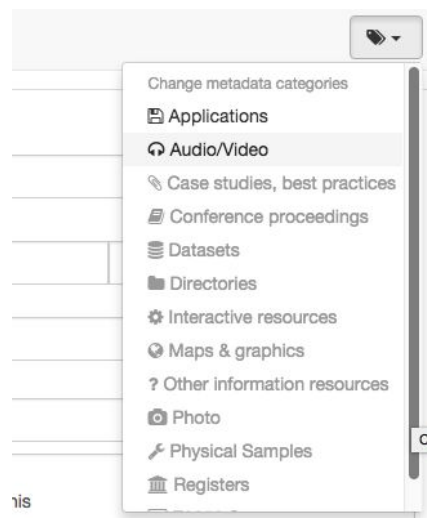
### Open Archive Initiative (OAI-PMH) Provider

<b>Datasearch</b>	<div style="border: 1px solid #ccc; padding: 2px;">Temporal extent</div> <div style="font-size: 0.8em; margin-top: 5px;">           Influences the behaviour of OAI when parameters from or until are supplied. If set to temporal extent, the temporal extent of the date described by the metadata is searched. If set to modification date, the date at which the metadata was last modified is searched. Please note: restart GeoNetwork to make this change active.         </div>
<b>ResumptionToken Timeout</b>	<div style="border: 1px solid #ccc; padding: 2px;">3600</div> <div style="font-size: 0.8em; margin-top: 5px;">           The number of seconds a OAI resultset stays in the cache and can be scrolled using the OAI resumptionToken. Please note: restart GeoNetwork to make this change active.         </div>
<b>Cachesize</b>	<div style="border: 1px solid #ccc; padding: 2px;">60</div> <div style="font-size: 0.8em; margin-top: 5px;">           The number of OAI resultSets that can be stored in the Cache.         </div>
<b>Maximum records</b>	<div style="border: 1px solid #ccc; padding: 2px;">10</div> <div style="font-size: 0.8em; margin-top: 5px;">           Number of maximum records to return in OAI responses         </div>

## 2.11. MM3 Metadata Categories and Category based harvesting

All the improvements about metadata categories are already available on the official GeoNetwork implementation.

To assign a category to a metadata, the easiest way is to go to the metadata editor and use the top menu to select the category.



Now we can also define a default category by group. This means, that all the metadata created on that group will have that category assigned by default. We can also force a metadata to use only the categories allowed for the group that owns the metadata. This option is disabled by default, if we want to filter the categories based on the group, we have to select the “Enable Allowed Categories” option.

All this category configuration about groups are handled on the administration console, on the **Users and Groups** section, where an administrator user can create, edit and remove groups from GeoNetwork.

**Category** Directories ▼

Default category for metadata created in this group

**Email**

Email for receiving notification

**Website**

Website containing information about the group.

**Logo**

Select from existing logos

▼

Upload

+ Choose or drop resource here

**Enable Allowed Categories** ☒

**Allowed Categories**

Applications  
Audio/Video  
Case studies, best practices  
Datasets  
Directories  
Interactive resources  
Maps & graphics  
Other information resources  
Photo  
Physical Samples

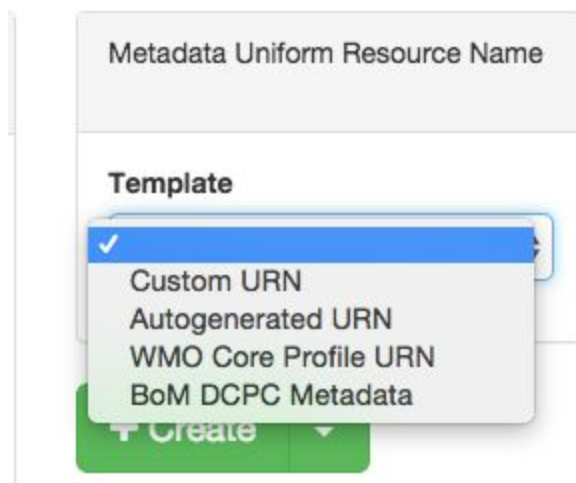


## 3. User's manual

### 3.1. Create metadata

The create metadata page allows to select the metadata template, group owner and define the metadata identifier.

For the metadata identifier, the following options are available:



The screenshot shows a web form titled 'Metadata Uniform Resource Name'. Below the title is a section labeled 'Template' with a dropdown menu. The dropdown menu is open, showing four options: 'Custom URN' (selected with a checkmark), 'Autogenerated URN', 'WMO Core Profile URN', and 'BoM DCPC Metadata'. Below the dropdown menu is a green button labeled 'Create'.

Custom URN: free-text metadata identifier.



The screenshot shows the 'Metadata Uniform Resource Name' form. The 'Template' dropdown menu is set to 'Custom URN'. Below the dropdown menu is a text input field labeled 'urn'.

Autogenerated URN: a random uuid is assigned to the metadata.

Metadata Uniform Resource Name

**Template**  
Autogenerated URN

Custom metadata identifier templates: a list of the metadata identifiers templates. When the user selects one of these templates, the form request the different values defined in the template.

Metadata Uniform Resource Name

**Template**  
WMO Core Profile URN

urn:x-wmo:md:int.wmo.wis:: TT AA II CCCC

**TT**

**AA**

**II**

**CCCC**

### 3.2. UI1 Subscription Management

Any user can browse all the subscriptions currently available for him. This is done clicking on his profile name, which will redirect the user to the **Users and Groups** section of the administration console. Then, on **My Subscriptions** tab, a table will be shown with all the currently available downloads. Each row have four options:

- View metadata
- Edit subscription
- Pause/Resume subscription
- Discard subscription

The functionality is the same as in [2.6. ADMIN5 Browse Subscriptions](#).

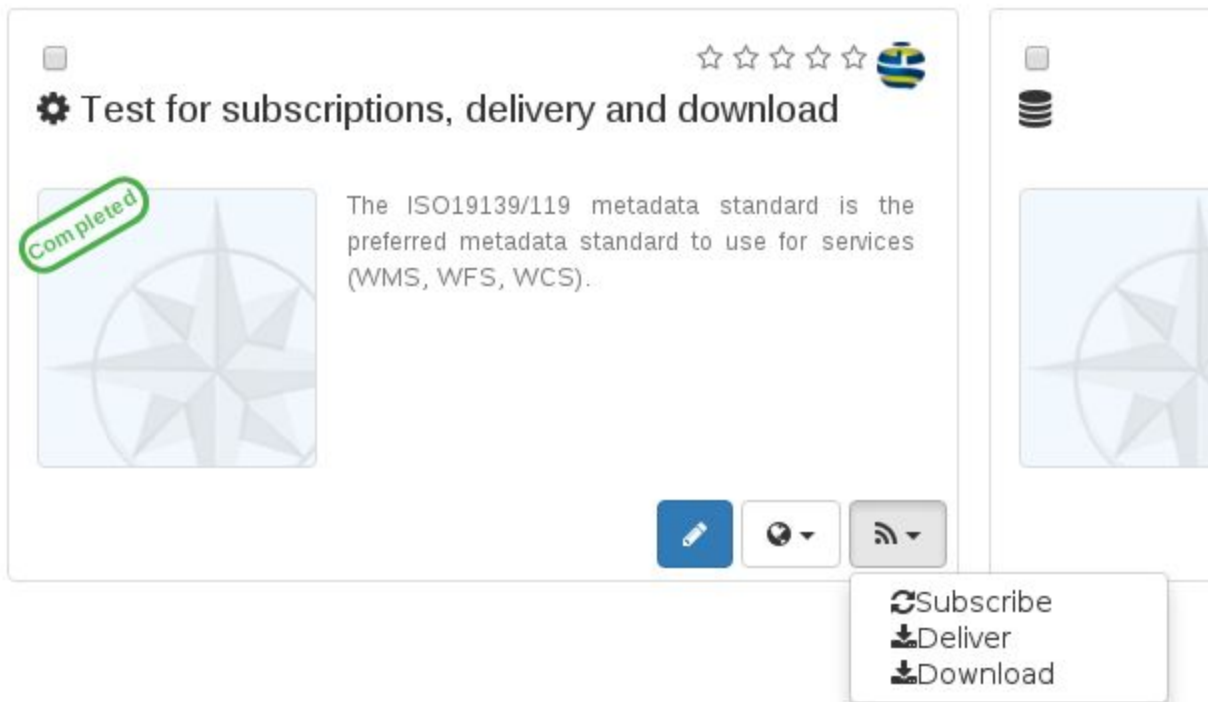
### 3.3. UI2 My Downloads Page

A user can check all the downloads currently available for his user. This is done clicking on his profile name, which will redirect the user to the **Users and Groups** section of the administration console. Then, on **My requests** tab, a table will be shown with all the currently available downloads. Each row have four options:

- View metadata
- Edit download
- Discard download
- (Optional) Link to download the product

### 3.4. DSS

GeoNetwork has been extended to show three new buttons both in the search results items and on the metadata view. This three buttons will only be shown if the OpenWIS service offers functionality to the metadata associated.




The buttons are:

- Subscribe
- Deliver
- Download

All three options will open a new window to configure it.

### 3.4.1. New Subscription

When a user clicks on the Subscribe button, GeoNetwork will show a modal dialog window with the dissemination options. Once you configure the primary dissemination (and optionally the secondary dissemination) desired and checked on the summary that the data is right, the subscription can be saved on the OpenWIS SOAP service.

 Send a request
 ×

Sub-selection parameters
Primary dissemination
Secondary dissemination
Summary

MSS/MFS
Public Dissemination

Compression

FTP

Host\*
Path\*
User\*
Password\*
☐ Show password

Advanced

Mail

Staging Post

✕ Cancel

◀ Previous

Next ▶

✓ Save

### 3.4.2. Ad-hoc request for delivery

GeoNetwork will show a modal dialog very similar to the previous one, but instead of creating a subscription on the OpenWIS service, it will request a single request for delivery based on the primary and secondary dissemination defined.

### 3.4.3. Direct Download via HTTP

The modal dialog opened will look very similar to the previous dialog, except that it only shows the primary dissemination and allows only the Staging Post option.