Data Hub System

Software User Manual

ISSUE: 1
REVISION: 1

AUTHORS: *ADRIANA GRAZIA CASTRIOTTA, BARBARA BORGIA*

serco

Table of Content

This document has been prepared in the frame of the contract 'Copernicus PDGS Technical Management Support', number 22204/09/I-OL, between ESA and Serco SPA

reviewed by: Guido Vingione *authorized by: Andrea Tesseri*

Introduction

This document applies to the DHuS system software having version number following the 0.9.0-2 that has been released on 16/03/2016. In the following, this version will be called "reference version".

Purpose and Scope

The purpose of this document is to describe the intended uses of the DHuS software and to provide a comprehensive step-by-step guide to the targeted user typologies.
In particular it describes (and dedicates a specific section to):

- How to install the software;
- How to administer, manage and operate the installed DHuS instance;
- How to make use of the DHuS capabilities offered by an installed DHuS instance.

The target audience of this document, therefore, includes:

- System administrators that will install the DHuS software and manage the DHuS;
- Final users that will benefit from the DHuS functions for their own purposes.

DHuS installation (and management) is normally of interest to Data Providers (which may want to provide access to their own data via the DHuS).
The actual, final users of the DHuS, instead, will either interact with the system via the provided Web-based Graphical User Interface (in which case they will not need to install any specific software component, nor to stick to any system requirement), or exploit the exposed machine-to-machine interface from their own software applications (in this case they will be application developers with programming skills).

Document Organization

Having in mind the target audience described above, this document is organized as follows:

-

- *Introduction:* Describes the purpose of this document, its target audience and its structure;
- *System Overview:* Describes the major DHuS capabilities and provides a high level description of its main components;
- *Installation Guide:* Provides step-by step instructions aimed at installing and setting up the system. This section is especially targeted to System administrators, that will install a DHuS instance whenever this is required.
- *Administration Guide:* Includes the DHuS management guidelines and is addressed to System Administrators and/or operators, who will have to make sure that the installed DHuS instance remains up and fully functioning and properly serves the needs of the final users.
- *User Manual:* Describes how the DHuS capabilities can be accessed and exploited by the "final users". It includes, in turn, two main sections: one describing the interactions between the human user and the available Web-based GUI and another one describing the machine-to-machine interface that can be exploited in external applications.
- *References:* Lists the documents applicable to this manual
- *Abbreviations and acronyms:* Expands the acronyms used in this document
- *Definitions:* Explains the meaning of specific technical terms used in this document

System Overview

The Data Hub Software (DhuS) is open source software developed by a Serco/Gael consortium to the purpose of supporting the ESA Copernicus data access.

The DHuS provides a simple web interface to allow interactive data discovery and download, and a powerful Application Programming Interface (API) that allows users to access the data via computer programs/scripts thereby automating/integrating the download within their workflow.

Different instances of the Data Hub are currently operated by ESA allowing for tailored managed levels of service:

- Research and General Public ('Sentinels Scientific Data Hub')
- Copernicus Service ('Copernicus Service Project Data Hub')
- International Partners ('Sentinels International Data Hub')
- Collaborative Ground Segment ('Sentinels Collaborative Data Hub')

The major functionalities of the Data Hub Software are schematically represented in figure below.

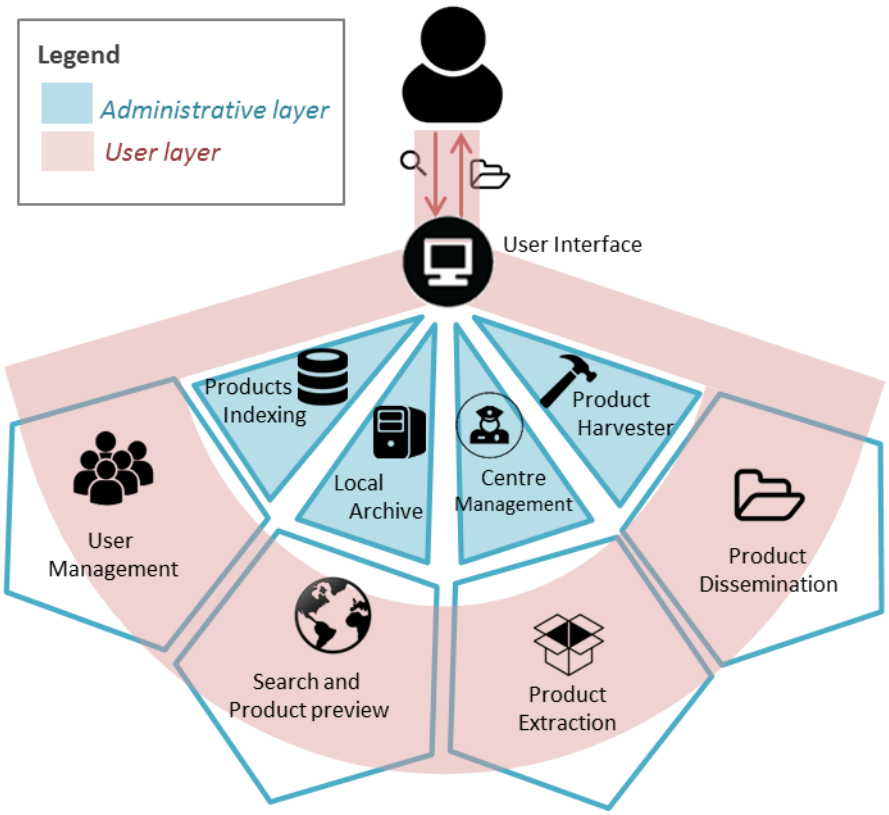

Figure 1: Data Hub Software functions

Product Indexing

The indexing module allows to reference products within the local archive, in addition the indexing service can reference products hosted in other DHuS instances.
The indexing is based on the creation of configurable metadata. The following set is considered mandatory when applicable for a product type:

- Content Date (e.g. sensing start, stop or data validity date)
- Content Geometry (e.g. footprint)

A more complete set of metadata is being defined considering the HMA specifications. In particular, the native metadata definition, from the product definition, will be "aliased" to a standard term from the HMA model, to facilitate a standard set of queries across data product types. Additional product-specific metadata, including "free text" attributes, will improve the usability of the Data Hub for all classes of users.

Local Archive

The local archive is the core element of the system; it retains the products managed at a given instance. The local archive is managed as a rolling archive. The eviction strategies supported by the referenced DhuS version is based on the ingestion time: the time interval following data ingestion is configurable.
Eviction can be triggered whenever the rules are met or according to higher-level strategy, such as the amount of local disk space usage. Eviction can be modulated according to the monitored usage of a product. To improve product download performances and enable fast product access, with potential processing at the same time, the local archive manages products in both compressed and uncompressed versions.

Centre management

The centre management service allows the unambiguous identification of the centre to distinguish it from other DHuS centres. It is used to apply the particular configuration of the local services as well as the configuration of its relationship, if any, with other centres.
Except for exceptional cases, the centre configuration modification shall not lead to any centre service interruption.

Product harvesting

The product harvester is the service responsible for the ingestion of external products into the local archive. The ingestion service defines an interface to allow definition of location, product type and characteristics for ingestion into the local archive. Whenever new products are ingested, the ingestion process registers an event with the dispatcher service, to inform it about the availability of the newly ingested products and to allow the propagation of

metadata to other potentially interested centres. An integrity measure is calculated on product ingestion that may be checked later on to ensure the product integrity.

User Interface

This module is in charge of providing the user with an interface for the discovery, visualization and downloading of products. It consists of two interfaces: a Graphical User Interface (modern and easy-to-use web application) and an Application Programming Interface (useful and mainly used for batch scripting, machine to machine scripts).

User management

This module is in charge of managing the user accounts for access to the DHuS. It allows the definition of roles and permissions of the users. Permissions are usually defined in terms of macro functions e.g. the ability to upload data, download data, view statistics, perform administration etc. This module is used to apply the user shared quota allocation schemes for restricting the download bandwidth.
This user account management service is configurable per DHuS instance to allow for "self-registration" at a hub or for "operator driven registration". Each instance can be configured differently for usage by a known community.

Search and product preview

This module is in charge to provide users the possibility to perform search via standardized API protocols (OData and Open search) and via the graphical user interface. Moreover, in addition to the typical searches allowed by other EO dissemination service interfaces, the DHuS graphical user interface provides the free text search, allowing intuitive searches via keywords for the data required, region of interest, time window etc.
The DHuS also allows users to save their defined searches, allowing both the reuse of that set of search parameters at a later date and the possibility to request an email notification whenever a new product matching the parameters is ingested.
Following product metadata ingestion, the DHuS routinely applies "post-processing" operations allowing browse previews, data type oriented metadata visualization, following a search.

Product Extraction

The product extraction module manages the ability to disseminate / retrieve sub-components of products (with granularity smaller than the individual files). It is considered a unique and unprecedented capability that is of particular relevance for the voluminous data products managed in the Sentinels mission.

Product Dissemination

The dissemination module manages the product dissemination, retrieval and access by the user.
The dissemination service supports several protocols, including the product transformation service, which allows transformation of the data at the time of product retrieval. The dissemination system supports the web user interface as well as machine-to-machine application programmable interfaces. Support for common internet tools and standards, (e.g. download managers), is also within the scope of this service.

Installation Guide

The DHuS is a web application, running within a Java Virtual Machine. All its middleware components, such as database and application servers, run inside the JVM container.

In order to allow integration into a hosting environment, the application needs to be installed and configured having well in mind what are the external interfaces to be used.

The application needs to manage *two flows*:
- the incoming flow: during the products ingestion process, the DHuS SW picks up products (compressed or not) from a folder and move them into another folder. We will call those folders, respectively, "inbox" and "incoming".
- the outgoing flow (how external users can search and download published data): this can happen using http (Tomcat) and, in particular cases, also ftp (service started by DHuS) on some dedicated service ports.

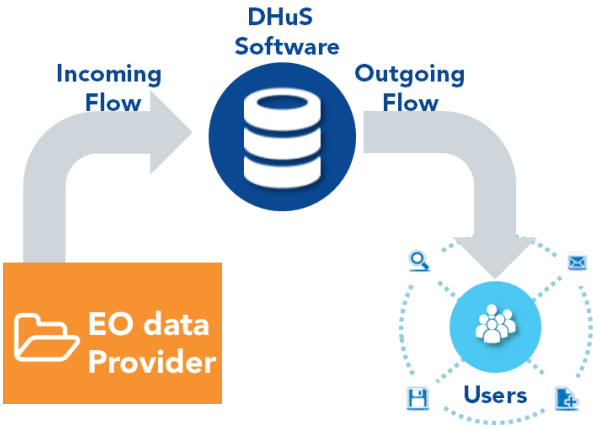

Figure 2 Principal DHuS Software Functionality

The *filesystems* used by the application, can be two or more;
- one filesystem is needed for storing the DB where the products are indexed, along with the logs and application binaries. This filesystem needs to reside on the local disks.
- one or more filesystems are used for archiving the products.

Given the volume of normal Sentinels production, it is also recommended to use an external disk for mounting the second filesystem, in order to cope with several TBs of products.

System Requirements

Hardware Requirements

The technical specifications of the DHuS are provided the following Table.

	Required performance		
	Minimum	**Medium**	**High**
CPU cores number	4-16	24	32
RAM	4-16 GB	32 GB	48 GB
Local disk	1 TB	1 TB	2 TB
Archive[1]	50 TB (1 month rolling archive)	200 TB (till 3 months rolling archive)	500 TB
Available External Bandwidth	100 Mbps	2 Gbps	10 Gbps
Internal Bandwidth	1 Gbps	4 Gbps	10 Gbps

Table 1: technical specifications

The Linux based operating systems in which the DHuS operability has been tested are:

- Debian 7.7
- Red Hat 6.7

Network Requirements

DHuS is accessed primarily via HTTP and FTP interface.
The Installation procedure of the DHuS SW must be performed using a **non-privileged user (not root)**; application installed in this way cannot start services listening on ports numbers smaller than 1024.

By default the HTTP interface is reachable on 8080 port that must be opened for inbound requests.
The DHuS FTP service is reachable, by default, on 2121.The DHuS provides also a mailing service based on an external SMTP server.
Following table describes the default DHuS network ports configuration:

Services	inbound	outbound
HTTP	8080	-
HTTPS	443	-
FTP	2121	-
SMTP	-	25

Table 2: Network ports configuration

Software Requirements

[1] Size of the archive is provided for a typical Copernicus production rate.

DHuS software is fully written in java and can be considered portable to any hardware platforms supported by JRE (Java Runtime Environment). The DHuS supports:

- all the Java JDK versions before the 7th (version 8 not yet supported)
- the Oracle distribution version 1.7.0_79.

It is recommended to use a Linux Operating System working on a multithread environment running in 64bit.

Java

The DHuS server requires Java Runtime Environment version 1.6+ being installed on the system.

Mailing service

An SMTP mail server should be made available to the DHuS System in order to allow its mailing functionalities.

Apache version

The proxy configuration is required in case the HTTPS protocol shall be used.
In this case, make sure the apache version compatible with DHuS is the number 2.2.15 with mod_proxy and mod_ssl.

The `httpd -v` command tells which config file Apache is using.

Downloading the Software Package

Data Hub System is developed by a Serco (http://serco.com/) and Gael Systems (http://www.gael.fr/) consortium under a contract with the European Space Agency (http://www.esa.int) in the frame of the Copernicus Programme. The SW code is made available according to the GNU Affero General Public License at https://github.com/SentinelDataHub/DataHubSystem.

Installation and Setup

Preliminary procedures

1. Create a user named 'dhus'. Every step in the installation procedure, if not explicitly mentioned, shall be performed as dhus user.
2. Create the installation folder

```
mkdir -p [installation-folder]
```

3. Download the DHuS package (shar package) and save it into the installation folder
4. Change the permissions on the file.

```
chmod +x dhus-XX.XX.XX.shar
```

5. Launch

```
./dhus-XX.XX.XX.shar
```

(the package will autoinstall).
Once executed, the system setting configuration file can be accessed and updated.

6. Once the autoinstall procedure is complete, create the following directories for the local archive, the incoming products, the database etc..:

```
Local archive →[install-dir]/data-local
Var →[install-dir]/var/
Incoming →[free_dir]/incoming
```

Note that the incoming and the Local archive shall be two different folders (e.g. **one cannot contain the other and vice versa**) not necessarily under the DHuS installation folder. Moreover they shall be located in a partition of the machine where there is a certain amount of space (more details would be specified in Table 1), especially for the incoming folder (the data managed by DHuS will be located here). The graph in Figure 3 depicts the purpose of the directories in the DHuS archive.

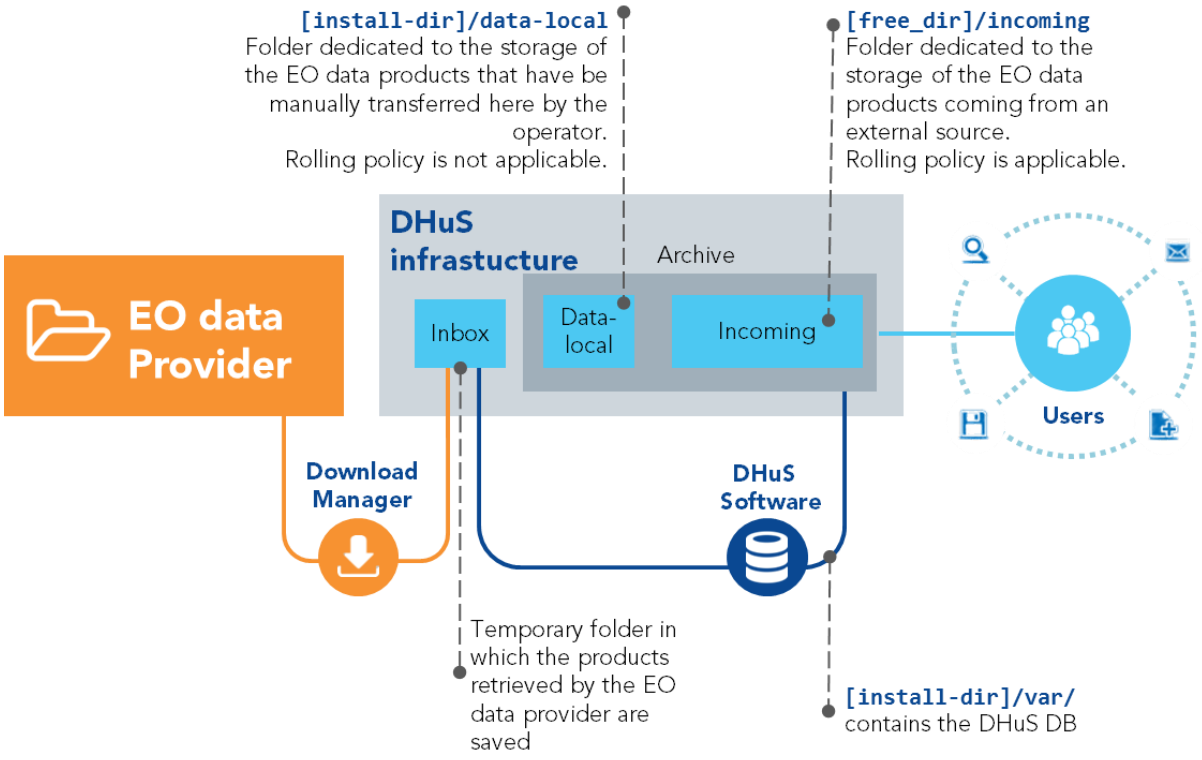

Figure 3: DHuS directories objectives

Software Configuration Manual

DHuS configuration files are contained in the `etc` folder created after the launch of the .shar installation package:

- Start.sh
- dhus.xml
- server.xml
- log4j2.xml

start.sh

The start.sh script contains the single startup command for the DHuS. Optional parameters to setup are:

```
-XX:MaxPermSize=<nn>
```

Sets the size of the Permanent Generation area, where class files are kept. These are the result of compiled classes and jsp pages.

```
-Xms<nn>
```

Specifies the initial memory allocation pool for the Java Virtual Machine (JVM).
For a machine with 16g memory available it is recommended to set <nn>=5g
For a machine with 32g memory available it is recommended to set <nn>=10g

```
-Xmx<nn>
```

Specifies the maximum memory allocation pool size for the Java Virtual Machine (JVM).
For a machine with 16g memory available it is recommended to set <nn>=12g
For a machine with 32g memory available it is recommended to set <nn>=24g

```
-XX:+UseConcMarkSweepGC -XX:+CMSIncrementalMode
```

Enable incremental Java Garbage Collection, distributing GC continuously during Java execution rather than periodically start heavier FullGC

```
-Dhttp.proxyHost=<proxyHostIP> -Dhttp.proxyPort=<proxyHostPort>
```

In case DHuS node doesn't have access to Internet, this setting is needed to allow geo-locations searches into DHuS using an open proxy access

Optional configuration

The configuration of the following configuration items is not mandatory (some of them are not present in the default start.sh script and must be added manually).

```
-DArchive.check=true|false
```

Force archive check at the start up (can also be scheduled in dhus.xml, see below): it checks the coherence between the products listed in the DHuS DB and the products physically stored in the DHuS rolling archive.
`Default value: false`

```
-DArchive.forceReindex=true|false
```

Force a re-index of the DHuS DB and the Solr DB by using the products currently stored in the DHuS rolling archive.
`Default value: false`

```
-DArchive.forceReset=true|false
```

Resets Archive at DHuS start (DESTRUCTIVE: use with caution)

```
Default value: false
```

```
-DAction.record.inactive=true|false
```

This parameter is a java property which has been introduced to set up the statistics support to the DHuS (*true* enables the parameter, while *false* disable it.).

```
     -DArchive.incoming.relocate=false                      \
     -DArchive.incoming.relocate.path=/path/to/relocation   \
```

TBW

The following settings are necessary for DHuS application and the modification is not recommended:

```
-Duser.timezone=UTC

-Dcom.sun.media.jai.disableMediaLib=true

-Dsun.zip.disableMemoryMapping=true

-cp "etc:lib/javax.servlet-api-3.0.1.jar:lib/*" fr.gael.dhus.DHuS
```

Configuration done…and now?

Once you make sure that all the parameters are set correctly, allow start.sh script to be executable:

```
chmod ug+x start.sh
```

dhus.xml

The dhus.xml contents are organized in 7 groups and provides all comments needed to understand how to configure them. In the following table, a high level description of the groups is provided.

Groups	Description
Crons	All the settings contained in this group have two parameters: ■ active: defines if the cron is currently active or not ■ schedule: defines the schedule of the cron The cron pattern is defined as: Seconds Minutes Hours Day-of-month Month Day-of-week [Year] You can find more information on: http://www.quartz-scheduler.org/documentation/quartz-1.x/tutorials/TutorialLesson06
Messaging	Settings contained in this group are used to configure the way DHuS is sending

	email messages. These configuration settings are used only at first launch of the system, when database is created from scratch and the values are used to populate it. In case settings need to be modified, use Management Panel inside the application. Modifying settings in these files, once database has been created, has no effect on the configuration.
Network	Settings contained in this group are used to limit user's concurrent access to network resources. The configuration is divided in inbound and outbound sections and each section has a PriorityChannel and SelfRegisteredChannel. Normal users belong to the SelfRegisteredChannel channel. Several settings can regulate different restrictions to available network resources.
Products	Settings contained in this group regulate how products are ingested into DHuS.
Search	Settings contained in this group regulate how the search could be performed by users, e.g. the default number of results returned by an OData query
Server	A part of the DHuS server configurations are contained in this group, other server configuration items are included in the server.xml file
System	Settings contained in this group indicated the system information like the configuration of the root password, rolling policy, DHuS DB and rolling archive paths in filesystems, etc..

In the APPENDIX A, an example of dhus.xml file is provided.

server.xml

The `server.xml` file contains the Apache Tomcat configuration settings. Additional settings could be included following the information provided in https://tomcat.apache.org/tomcat-7.0-doc/config/http.html. If not explicitly specified, the DHuS will set the server default values.

Please note that:
<Connector port="8080"
indicates the http service port to setup: the default value is "8080"

<Host name="localhost"
This parameter indicates the server hostname.

In the APPENDIX A, an example of a sever.xml file is provided.

Log4j2.xml

The log4j2.xml contains the log settings. It is possible to raise or lower the log level as needed.
In the APPENDIX A, an example of a log4j2.xml file is provided.

Installation

Start the new DHuS version

```
nohup /bin/bash start.sh &> <install-dir>/logs/logs.txt &
```

User Interface Configuration Manual

This section deals with the configurability of the GUI which allows a wide set of configuration actions which do not need a restart of DHuS to be applied.
Due to the growth of the different centres and related installations, a new configuration management module has been added into the AJS web app. It allows configuring various aspects of the GUI; mainly it is related to style, texts and layout:

1. Title (shown in the header bar)
2. Sections visibility (Cart, Profile, Sign In)
3. URL and texts of the link logos (shown in the header panel)
4. Version text (shown in the info panel)
5. Data Hub Logo (shown in the info panel)
6. Mission Tag (shown in the Product List panel)
7. Mission footprint style and color (shown in the Map panel)
8. Advanced Search Mission specific fields (shown in Advanced Search Panel)
9. Map Layer (shown in the Map View)

Please note that all the settings are included in the client side (2 text files), thus it is possible to change a parameter without restarting the DHuS, but just doing a refresh via browser.

How to change a parameter?

The files in charge of the GUI configuration management are located in:

```
[install-dir]/var/tomcat/webapps/new/config
```

They are:

- appconfig.json (includes 1,2,3,4,5)
- styles.json (includes 6,7)

Advanced Search Configuration

A special attention goes to the configuration of the advanced search mission specific fields.

The configuration file appconfig.json has been updated in order to manage mission specific filters.

A "missions" section has been added, containing an array with the following structure:

"name": <label show for filter>, "indexname":
<solr_metadata_index_name_identifying_filter>,
"indexvalue":<solr_metadata_index_value_identifying_filter>, "filters":[filter_array]

where [filter_array] is an array of mission-specific filters with the following structure:

"indexname": <solr_metadata_index_name_identifying_filter>
"indexlabel": <label show for filter>
"regex": <regex_to_be_used_to_validate_the_filter_value_if_needed>
[OPTIONAL] "indexvalues": <list_of_all_the_accepted_values> (if present it appears a combobox containing the list of all specified values, otherwise nothing appears. present

Here below an example of filters configuration specific for S1 and S2 missions.

…

```
…
"missions": [
        {
            "name": "Mission: Sentinel-1",
            "indexname": "platformname",
            "indexvalue": "Sentinel-1",
            "filters": [
                {
                    "indexname": "producttype",
                    "indexlabel": "Product Type (SLC,GRD,OCN)",
                    "indexvalues": "SLC|GRD|OCN",
                    "regex": ".*"

                },
                {
                    "indexname": "polarisationmode",
                    "indexlabel": "Polarisation (e.g.HH,VV,HV,VH,...)",
                      "indexvalues": "HH|VV|HV|VH|HH+HV|VV+VH",
                    "regex": ".*"

                },
                {
                    "indexname": "sensoroperationalmode",
                    "indexlabel": "Sensor Mode (SM,IW,EW,WV)",
                    "indexvalues": "SM|IW|EW|WV",
                    "regex": ".*"

                },
                {
                    "indexname": "relativeorbitnumber",
                    "indexlabel": "Relative Orbit Number (from 1 to 175)",
                   "regex": "[1-9]|[1-9][0-9]|[1-9][0-7][0-5]"
               }
            ]
```

```
        },
        {
            "name": "Mission: Sentinel-2",
            "indexname": "platformname",
            "indexvalue": "Sentinel-2",
            "filters": [
                {
                    "indexname": "cloudcoverpercentage",
                    "indexlabel": "Cloud Cover % (e.g.[0 TO 9.4])"
                }
              ]
        }
    ]
```

Once you have changed a value in the file, you only need to refresh your browser to see the change immediately applied. **No need to restart the DHuS.**

Version Upgrade

Dependencies

This installation manual provides the upgrading DHuS version manual which means the installation of the reference version using a DB created during an installation of an older version of DHuS. The following instructions are ensured for all versions after the 0.4.3-1.
Here below the list of configuration changes present from 0.4.3-1 to the reference version. Unless explicitly mentioned, the version which includes the change in configuration parameter is reported in the "including version" column.

	change	including version
start.sh	■ keep default GC behavior even if it is not explicitly specified ■ introduced the JVM "-server" flag to increase performances **from 0.6.0 version** (See http://www.oracle.com/technetwork/java/whitepaper-135217.html#2 for details) ■ Added exhaustive list of the supported properties as comments. ■ remove unused properties: `-Daction.record.inactive=true` ■ increase the memory usage to 56 Gb (-Xms56g -Xmx56g)	≥ 0.5.5-2
dhus.xml	- support of passive port (default configuration `passivePort="30200-30220").` For further details see: https://mina.apache.org/ftpserver-project/configuration_passive_ports.html	≥ 0.6.1

	- the internal server configuration has been included in server.xml file so the line `<server protocol="http" host="localhost" port="8080">` substituted with `<server>` - The processing parameter in the system configuration group has been moved in the server.xml, so the line `<processing corePoolSize="4" maxPoolSize = "10" queueCapacity="10000" />` has been substituted with `<processing corePoolSize="4" />`	
server.xml	Some parameters in the "dhus.xml" configuration file have been removed and collected in this new (from 0.5.1 version) configuration file called "server.xml": ■ maxConnections can now be set in "server.xml" in Connector object ■ maxThreads can now be set in "server.xml" in Connector object ■ nio can now be set in "server.xml" in Connector object by setting protocol value to "org.apache.coyote.http11.Http11NioProtocol" ■ port can now be set in "server.xml" in Connector object ■ host is forced to "localhost", because tomcat is always deploying on localhost. ■ protocol is forced to "http", according to existing working version (currently setting https on local server is not working, meaning this is not used.) This allowed to set any other tomcat configuration parameter. The full server.xml configuration can be found here : https://tomcat.apache.org/tomcat-7.0-doc/config/ The full list of default tomcat parameter are provided here: https://tomcat.apache.org/tomcat-7.0-doc/config/http.html#Standard_Implementation, ■ from 0.6.0 version o the default connector port has been changed (from 8080 to 8081) o the keep alive time out parameter has been introduced ■ From 0.8.0 version, the server.xml file contains the <Valve> option enabling some logging roles.	≥ 0.5.1
synonyms.txt	A fixed list of countries is proposed instead of free text, in order to improve the statistics and avoid variants for one country. This file lists the ISO 3166-1 country names and their synonyms in the following format; `iso_country_name: synonym1, synonym2`	≥ 0.4.4
log4j2.xml	Removed the hardcoded absolute path to log file (use relative path instead)	≥ 0.6.1

Table 3: configuration file changes from 0.4.3-1 to reference version

DHuS version updating manual

Many aspects of DHuS first installation don't need to be repeated when upgrading application to a new release. In the following procedures the reference version will be called "*new_version*" and the older version, the version previously installed on the same instance, will be called "*old_version*"

1. Access to the chosen installation folder (/data is recommended) and create the installation directory:

```
mkdir dhus-<new_version>
```

(It's not necessary to touch the already present archive, the database is copied and then migrated at first start, so links to products remain intact and continue to point to the same archive)

2. Create the new layout:

```
mkdir -p dhus-<new_version>/logs
mkdir -p dhus-<new_version>/var
```

3. Change the execution permissions

```
chmod +x dhus-XX.XX.XX.shar
```

4. Launch

```
./dhus-XX.XX.XX.shar
```

5. Access the installation directory and rename all the *.sh files (produced with the installation) as *.sh.orig executing the command:
rename .sh .sh.orig *.sh

6. then access the /etc directory and rename all the *.xml files as *.xml.orig launching the same command as before:

```
cd /etc
rename .xml .xml.orig *.xml
```

7. check that all the .xml and .sh files are correctly renamed respectively as .xml.orig and .sh.orig.

8. copy all the .sh and all the .xml files, and synonyms.txt files from the folder of the previous version *(please note that synonyms.txt is not present in versions older than the version 0.4.4*):

```
cp -r dhus-<old_version>/*.sh dhus-<new_version>
cp -r dhus-<old_version>/etc/*.xml dhus-<new_version>/etc
cp -r dhus-<old_version>/etc/synonyms.txt dhus-<new_version>/etc
```

9. Change the configuration files depending on the <old-version> number (see Table 3 for details on configuration files changes). Example of the updating configuration procedure from 0.4.3-1 to the reference version is provided in the next section.

10. Check if an older DHuS version is running

```
ps -edf | grep java
```

if in the list of active PID, one of them is reporting the text of the start.sh file and it is running under dhus user permission, it means that the older version of DHuS is running.

11. If an older version of DHuS is running, stop it

```
dhus-<old_version>/stop.sh
```

12. Copy the Database and SolR folder from the previous version

```
cp -rp dhus-<old_version>/var/database dhus-<new_version>/var/
cp -rp dhus-<old_version>/var/solr dhus-<new_version>/var/
cp -rp dhus-<old_version>/var/tomcat dhus-<new_version>/var/
```

13. Change the var path in the dhus.xml and check if every path containing "&varFolder;/" path are still respected

```
 <!ENTITY varFolder "dhus-<new_version>/var/">
]>
```

14. Start the new DHuS version

```
nohup /bin/bash start.sh &> dhus-<new_version>/logs/logs.txt &
```

Example of configuration changes updating DHuS from 0.4.3-1 version to the reference version

Start.sh configuration

- Open `dhus-<new_version>/start.sh.orig` (that is the new one)
- Set the memory usage parameters according the own needs (the default values are `-Xms56g -Xmx56g`)
- Change the properties removal into false (to allow collection of statistical information)

```
    -Daction.record.inactive=false                    \
```

- Save start.sh.orig and remane it as start.sh

```
mv start.sh.orig start.sh
```

Dhus.xml configuration

- Open `dhus-<new_version>/etc/dhus.xml` (that is the old one)
- Change the following lines:

Original line	Changed line	Comments
`<server protocol="http" host="localhost" port="8080">`	`<server>`	
`<ftp port="2121" ftps="false" />`	`<ftp port="2121" ftps="false" passivePort="30200-30220" />`	The passivePort parameter can be configured after the update of the version
`<processing corePoolSize="4" maxPoolSize = "10" queueCapacity="10000" />`	`<processing corePoolSize="4" />`	corePoolSize parameter shall be configured according the own needs

- Add the following line at the end of the system group (after tomcat path configuration:

```
<executor enabled="false" batchModeEnabled="false" />
```

- Save `dhus.xml`

Server.xml configuration

- Open `dhus-<new_version>/etc/server.xml.orig` (that is the new one- the old one does not exist in version 0.4.3-1)
- Chech the configuration of tomcat parameters, especially the following line (in 0.4.3-1 version tose information were in the dhus.xml file)

```
<Connector port="…"
                …>
```

- Save server.xml.orig and remane it as server.xml

```
mv server.xml.orig server.xml
```

Starting and Stopping a DHuS Instance

Start DHuS

Once the DHuS files are configured as shown in Software Configuration section, execute, as dhus user, the following command in the folder where the DHuS is installed:

```
nohup /bin/bash start.sh &> [installation-dir]/logs.txt &
```

Stop DHuS

To stop DHuS, execute, as dhus user, the following command in the folder where the DHuS is installed:

```
/bin/bash stop.sh
```

Administration Guide

Login

Once the installation package (see Installation Guide chapter) has been successfully installed, the DHuS server can be accessed online (https://dhus.xxx.zz) or on local URL (https://localhost/).
The Administrator functionalities are accessible by the GUI at https://dhus.xxx.zz/new/#/home after the login as root, using the default settings. The button is displayed in the upper right side of the DHuS Home page.

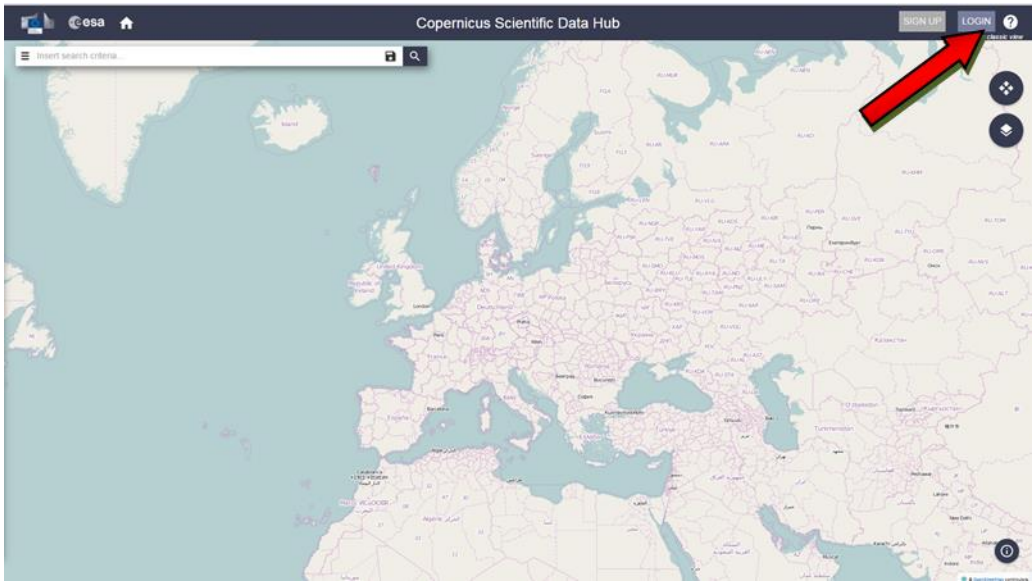

Figure 4: GUI login

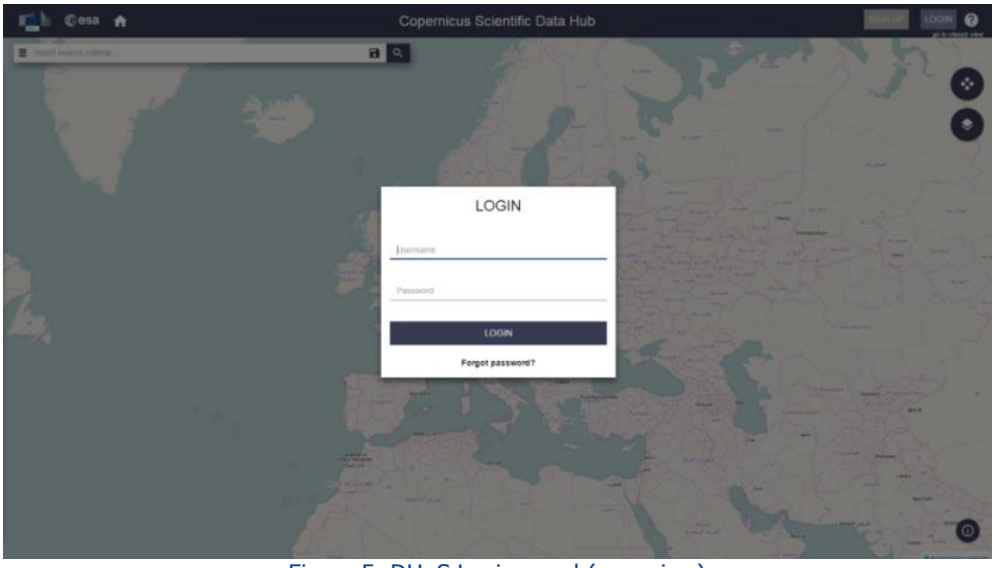

Figure 5: DHuS Login panel (user view)

Panels description

The DHuS provides the Administrator a series of panels to fulfil every service. We report here how to access them using the GUI.
The list of panel is provided here below:

- Search panel
- Upload panel
- Profile panel
- Cart panel
- Management panel:
  - Users
  - Collections
  - System
  - Eviction
- OData synchronizers panel

Once the administrator has logged in, the panels are accessible clicking on the "user icon" on the upper right side of the page.

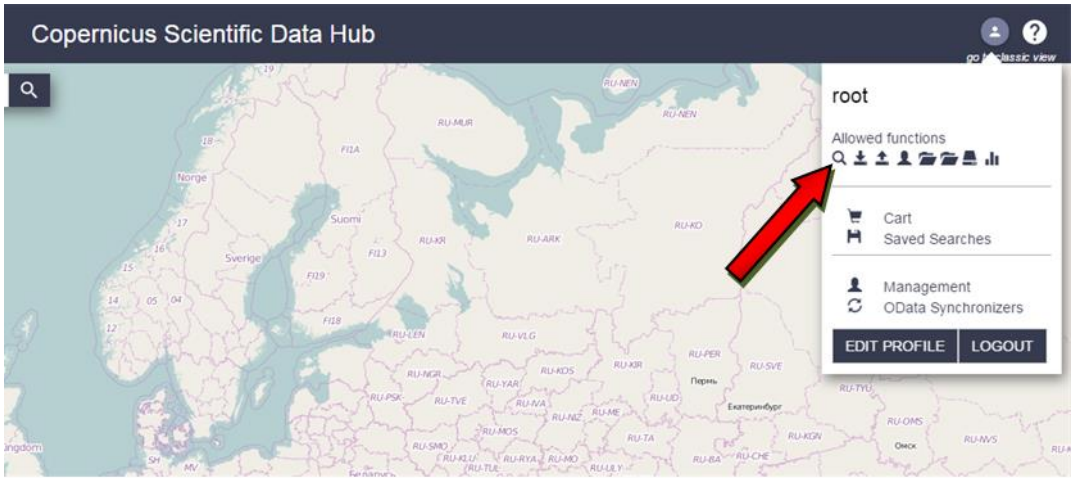

Figure 6: GUI administration panels

Products Upload

The Upload feature is available only to the administrator. DHuS system makes available an incoming space to let the user upload a product. Once uploaded, data is processed to be referenced by DHuS clients.
This panel gathers all the information necessary to perform the upload (at least the path to the product).
Optional: Assignation of a product to a collection is manually set by the uploader. A product can be included in any collection.

The DHuS allows the ingestion of Sentinels products using 4 methods:

- Ad hoc upload
- Creating a file scanner

- Synchronizing remote archive

Ad hoc upload	File scanner	Synchronizing remote archive
⊕ HTTPS product transfer ⊕ Allowed collection targeting ⊖ No cumulative uploads ⊖ Only zip format compatible	⊕ Cumulative uploads ⊕ Parallel uploads ⊕ Optional FTP transfer protocol ⊕ SAFE & zip format compatible ⊕ Allowed collection targeting	⊕ Cumulative FAST upload ⊕ Parallel upload ⊕ HTTP/HTTPS transfer protocol ⊕ Product transfer not necessary ⊕ Allowed collection targeting ⊖ Other independent DHuS instance needed

Table 4: products upload methods

The first two methods are accessible by the upload panel, the third is accessible via a dedicated odata synchronizer panel:

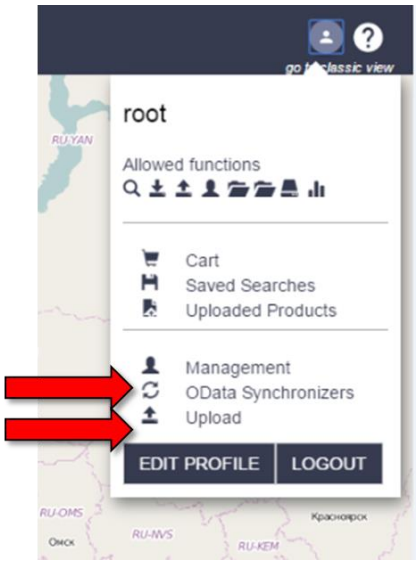

Figure 7: Access to upload features.

Ad hoc upload

Once in the Upload panel, it is possible to perform the upload of a product: select the input products, then (optional) select a collection in the list of collections and click on the "Upload" button. The upload will start and at the end of it, a pop up will notify that the upload is over.

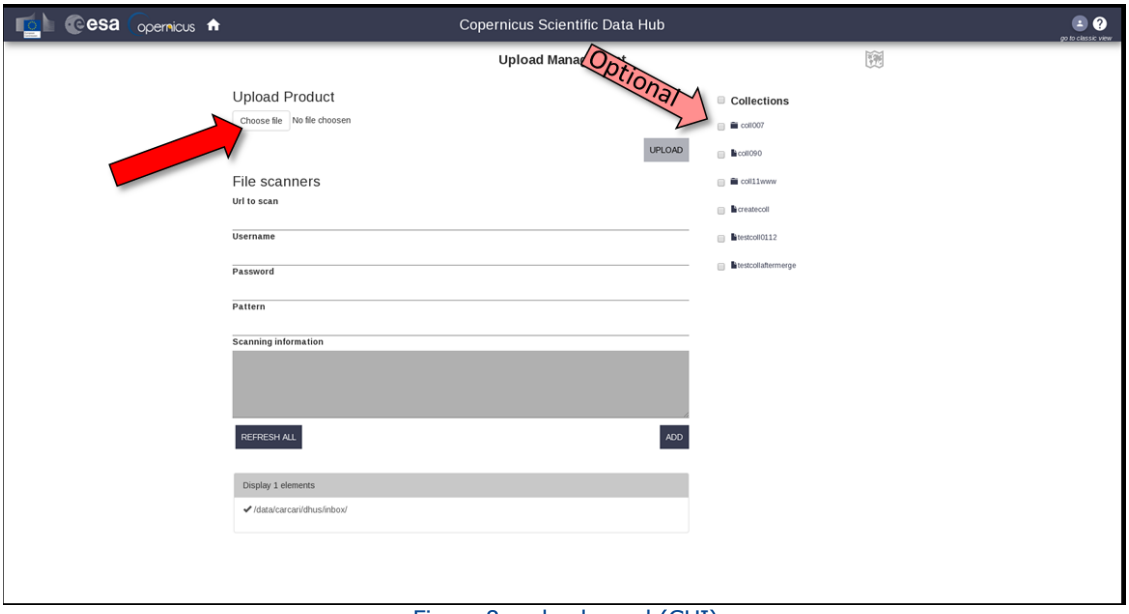

Figure 8: upload panel (GUI)

Upload via file scanner

If the upload has to be periodic, a scanner can be configured with the panel highlighted by the red arrow in Figure 9.

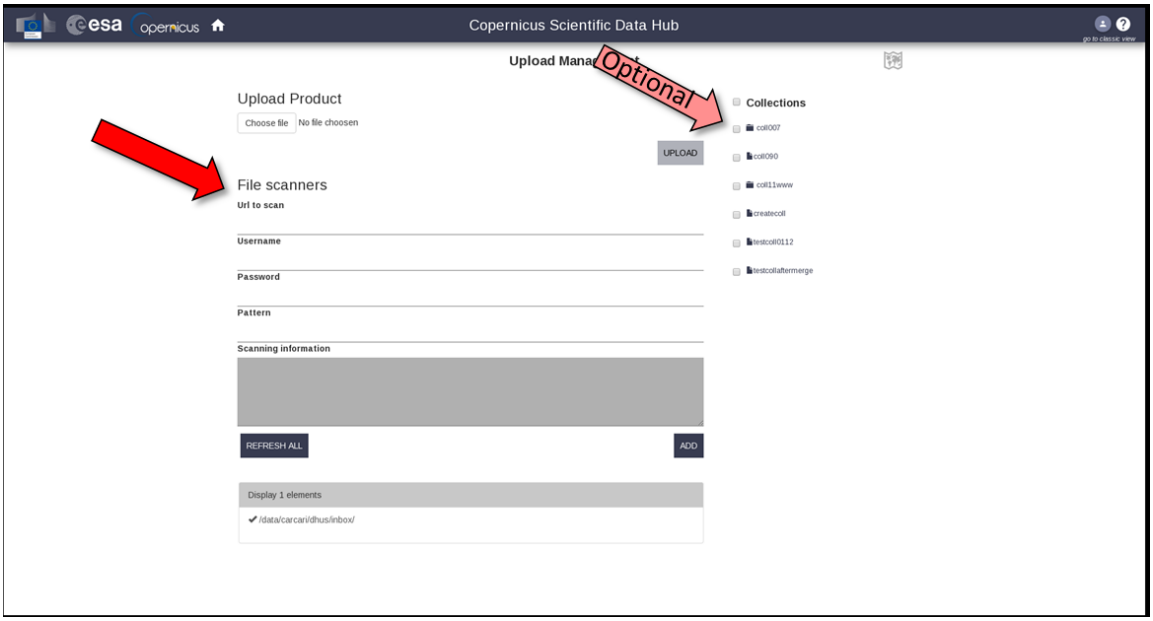

Figure 9: file scanner panel (GUI)

To create a file scanner

- Access the upload panel
- Fill the Url to scan field with the path of the folder containing the products (if the products are in the same machine where DHuS is installed, the field shall be filled as 'file:///path/of/the/folder').

- If the products are located on an external data provider (accessible via ftp), configure the username and password to access the machine; otherwise the username and passwords will not be necessary.
- To upload just specific types of product, configure the 'Pattern' field according to the regular expression roles explained in http://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html (e.g. `"S1[AB]_\p{Upper}{2}_(SLC|GRDM).*"` to upload only the SLC and the GRDM products)
- Click on the 'add' button. In the lower part of the page it will be written 'when' the file scanner will be activated again.

Upload synchronizing remote archive

The DHuS allows synchronizing products from another DHuS instance. For further details, go to OData Synchronizers panel section.

DHuS Administration

The DHuS provides the Management panel and it contains 4 subpanels called

- Users
- Collections
- System
- Eviction

Figure 10: management subpanels (GUI)

Here follows a brief tutorial for using the management panels via the GUI.

User Management Panel

The administrator management panel allows managing users. This means that the administrator can create, edit and delete any user.

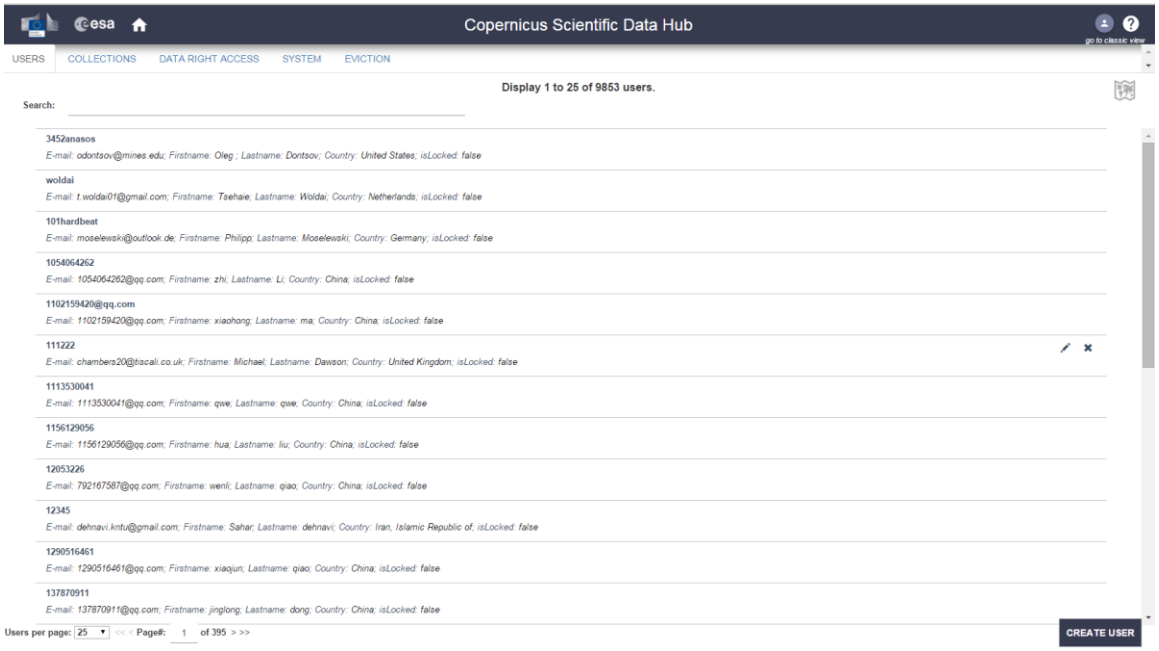

Figure 11: DHuS User Management panel (GUI)

DHuS implements a user management system that prevents uncontrolled accesses and manipulations from unauthorized users. DHuS proposes a user authentication and authorization strategy defined in its internal Database. Users are able to register or sign-in and the administrators are able to configure the user/group permissions from the Web user interface

The user management activities are:

1. to create or delete a user;
2. to authorize the user to access a list of services;
3. to update a user profile;

How to register a new user?

The Administrator shall:

- Access the DHuS page
- Perform the login
- Select the Management Panel and then select the Users management panel
- Click on the "Create" button in the lower part of the User management, which will enable the form here below

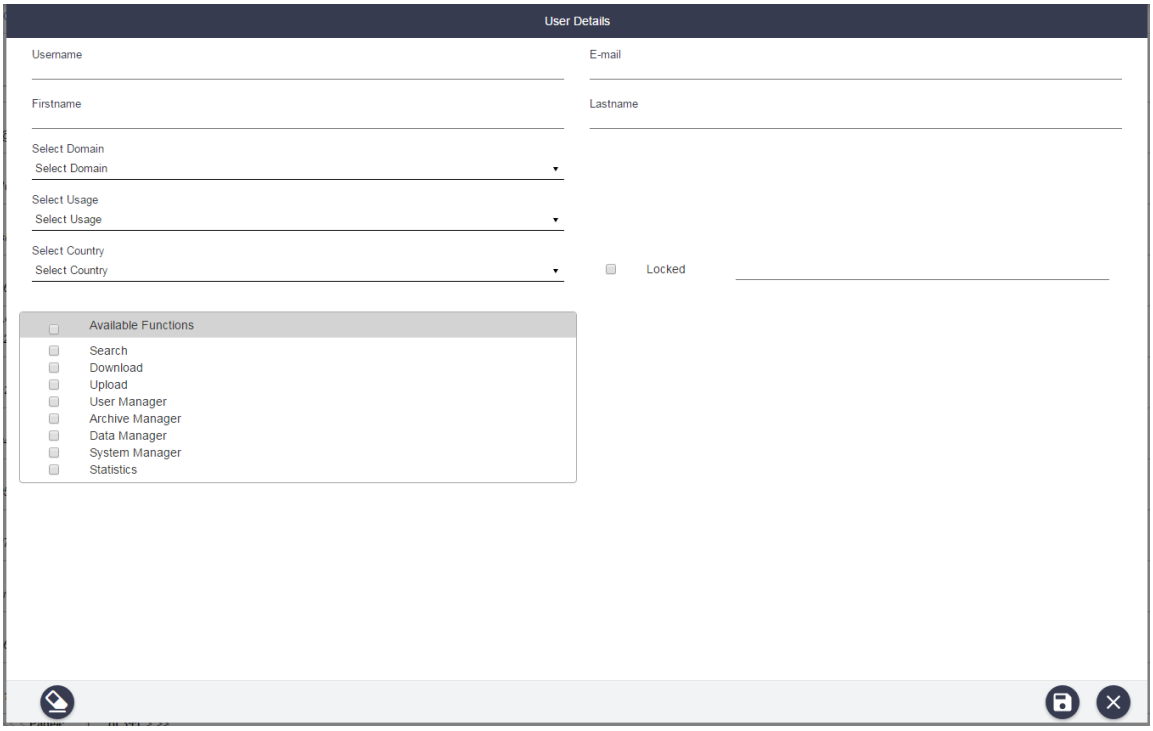

Figure 12: User creation form (GUI)

- Fill in the new user creation form (note that the fields marked with an asterisk are mandatory) and click on the functions that the user shall be able to use
- Click on the "save" button to complete
- Then the email notification service will send an e-mail to the user with his profile information (login, first name, last name, available services...) including a generated password.

The administrator has the possibility to modify user's authorization settings and information. To modify whatever authorization setting or user information, the Administrator, before executing the following "how to" procedures, has to:

- Access the DHuS page
- Perform the login
- Select the Management Panel
- Select the Users Management Panel
- Select the name of the user in the users list on the left side of the user management panel

How to lock the selected user?

Click on the "locked" checkbox under the Registration form in the right side of the panel,

- The administrator shall also indicate the reason of this locking process in the box on the right,
- Click on the "save" button to complete,
- Then the email notification service will send an e-mail to the user with his profile information (login, first name, last name, available services...) including locking notification and its relative reason, if it has been indicated.

How to delete the selected user?

- Click on the "Delete" button to delete (note that clicking on "Cancel" will just cancel any changes made).

Figure 13: Update and delete users (GUI)

- The email notification service will send an e-mail to the deleted user with the communication of the deletion process.

Collection Management Panel

Products are gathered into collections. Collections management consists of:

1. creating or deleting collections;
2. adding a sub collection or a collection parent.

The Collection management panel also lists a set of products to be attached to the collection. The selection of collections is possible by browsing the collection hierarchy on the left. To access the collection management panel, the Administrator has to click on the "collections" link, sited in the upper left side of the management panel.
The collection management panel here below will open. It contains the list of collections on the left and the list of archived products on the right.

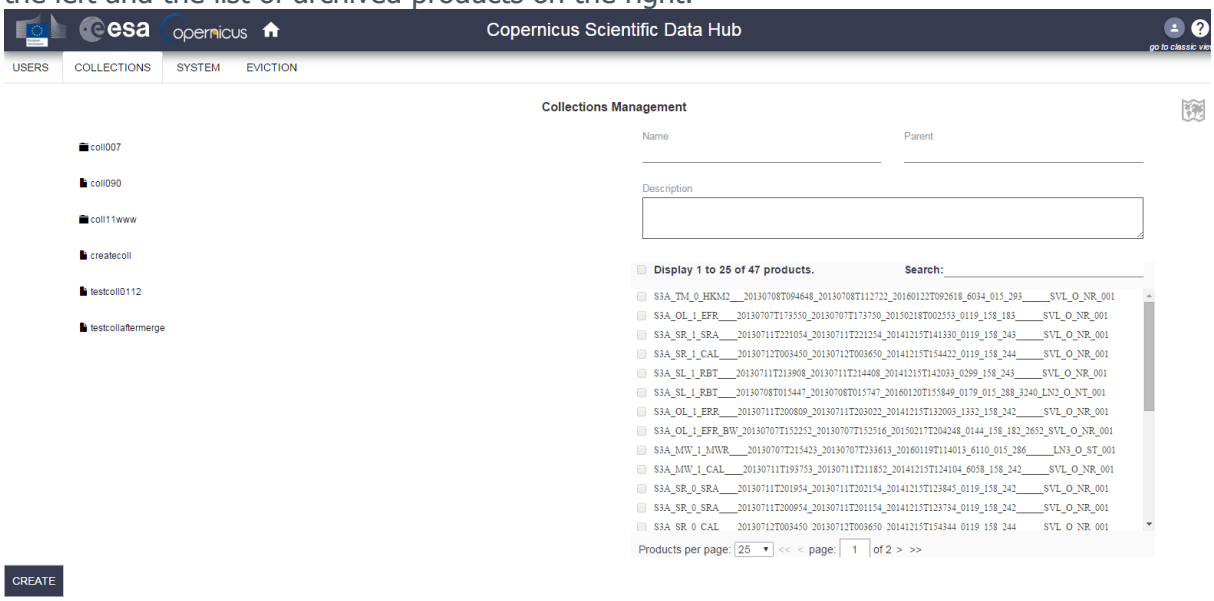

Figure 14: collection management panel (GUI)

The administrator can manage the collection: he can create new collection/subcollection and delete an existing collection/subcollection,

In the following subsections some "How to" tutorials are presented; the steps described in any of these tutorials can only be performed after the following preliminary actions:

- Access the DHuS page,
- Perform the login,
- Go to the collection management panel.

How to create a new collection?

- Click on the "create" button in the collection management panel;
- Insert the collection information in the upper right side of the panel (the name of collection is mandatory),
- (optional) select (by clicking on the associated check box) the products to be added to the collection,
- Click on the "save" button to register the new collection or click on the "cancel" button to abort the 'creation of collection' procedure.

How to create new sub collection?

- Click on the collection inside, which creates the new sub collection,
- Click on the "sub collection" button in the collection management panel,
- Insert the collection information in the upper right side of the panel (the name of collection is mandatory),
- Click on the "save" button to register the new collection or click on the "cancel" button to abort the 'creation of collection' procedure (note that clicking on "delete" will delete the collection in which you wanted to create the sub collection).

How to delete a collection/sub collection?:

- Click on the collection/sub collection to delete;
- Click on the "delete" button,

Note that the collection management page includes a **searching box**. It is useful to know if a product is collected somewhere.

System Management Panel

The system management is used to configure basic information in the system.
The Administrator from here can access the following sections:

1. Mail configuration: In this form it is possible to configure the SMTP server address, the username, password and e-mail account details to send communications to the users.

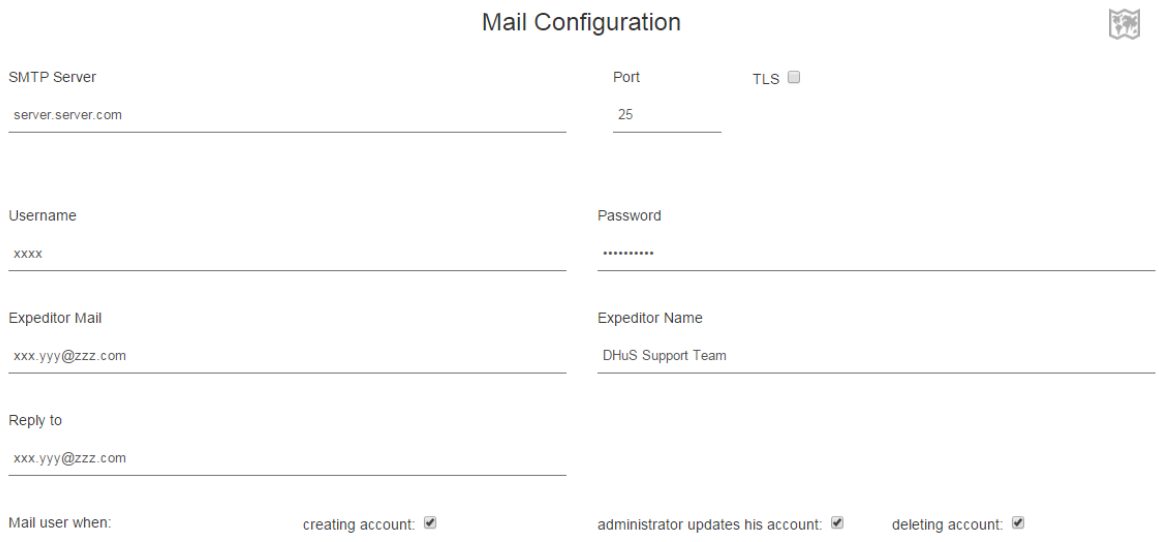

Figure 15: Mail configuration management panel

2. Support information: For any support information it is possible to contact the DHuS Support Team sending an e-mail to dhus@xxx.zz.

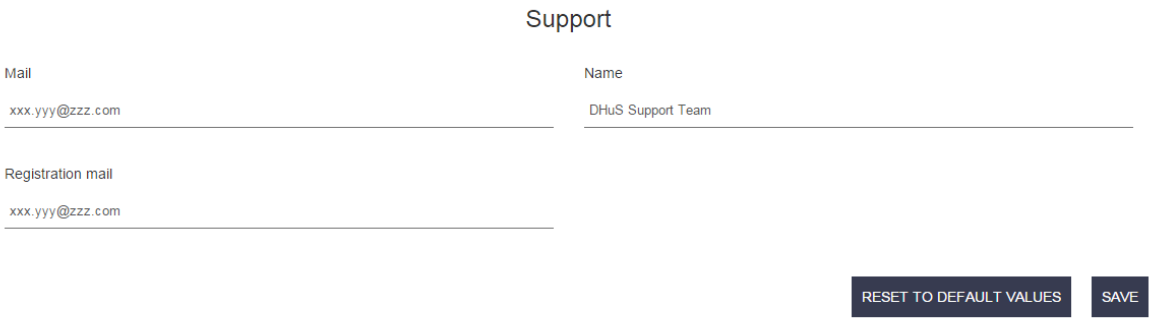

Figure 16: Support configuration management panel

3. Root configuration: from this panel it is possible to change the administrator password. To do so, insert the old password, the new one and then confirm the new password.

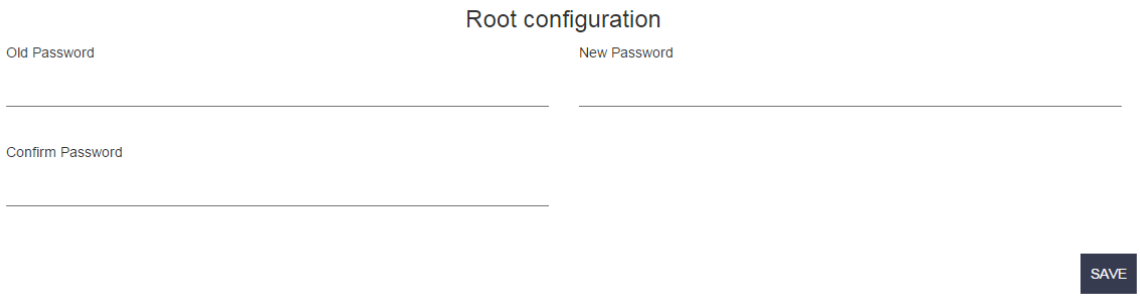

Figure 17: Root configuration management panel

4. **Restore database**: in the dhus.xml file it is possible to configure DHuS so that it performs a periodical dump of the database. From this panel it is possible to restore the database dump.

Figure 18: Restore database panel

To do so, perform the following steps:

- Click on the drop-down menu in the `restore-database' section: the list of available dumps will be displayed through a list of dates (date during which the dumps have been performed).
- Select the desired date and then click on "restore". DHuS will automatically stop and restart. Once DHuS will be up again, it will contain just the data inserted before the selected dump date.

5. **Synchronize Local archive:** obsolete function, do not use;

Eviction Management Panel

The Data Eviction Service is responsible for removing data to keep to the Data Store sizing constraints. The maximum occupied space for each archive depends on the configuration. The administrator can handle the eviction of products through the Eviction panel here below.

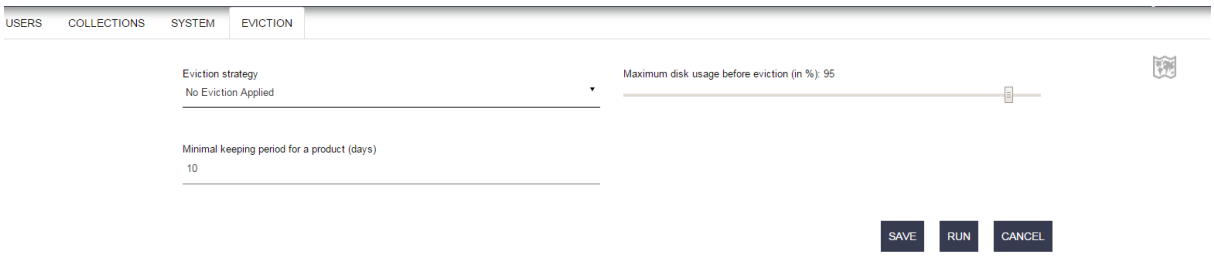

Figure 19: Eviction management panel (GUI)

The eviction rules are:

1. First In First Out (FIFO);
2. Least Recently Used (LRU).

They can be chose through the drop-down menu named "Eviction strategy".
LFU and LRU are defined using a system of "hit points", calculated with the number of searches and downloads for each product.
The service will log (in the panel on the lower side of the page) any evicted file in the Database and flag the product/entry by removing/updating its Data Store URL that is no longer relevant.

How to activate the archive rolling policy?

In order to activate the eviction, perform the following steps:

- Access the DHuS page
- Perform the login
- Select the Management Panel and then select the Eviction management panel
- Select the Eviction strategy using the drop-down menu
- Configure the "Maximum disk usage before eviction" depending on how much of the machine space can be occupied by data before triggering the eviction (e.g. if the parameter is set to 80, when the disk will be full at 80%, the eviction will be automatically activated)
- Configure the "Minimal keeping period for a product" parameter. This parameter represents the number of days each product will be kept in the DHuS archive before being evicted (e.g if the parameter is set to 3, the eviction will delete all the products present in the archive for more than three days.)

OData Synchronizers panel

The OData synchronizers panel is available just in the GUI. The DHuS provides end users an OData synchronizer service able to populate a DHuS instance with the data stored on the rolling archive of another DHuS instance. The DHuS instance that contains the data to be synchronized is called "back end" instance, while the one that shall receive the data is called "front end" instance.
In case the rolling archive of the BE contains some products that are not present in the FE, once the synchronization is running, the metadata of the products present in the BE instance that are not in the database of the FE instance will be mirrored.

Preconditions

The FE/BE instances should be configured as follows:

- BE: DHuS instance with no quota limitation and having a user with the 'archive management' function enabled.
- FE: having the synchronization functionality enabled, meaning that the dhus.xml of the FE shall contain the following setting:
  <executor enabled="true" batchModeEnabled="false" />
- BE and FE shall have the incoming folders in the same filesystem

The OData Synchronizers panel allows the creation and update of synchronizers among two or more DHuS instances.

How to create a new Synchronizers?

The Administrator shall

- Log in as Root in the front end DHuS instance and select the tab "user profile"
- Select the panel "OData synchronizers" (see Figure 7)
- Click on "Create synchronizer"

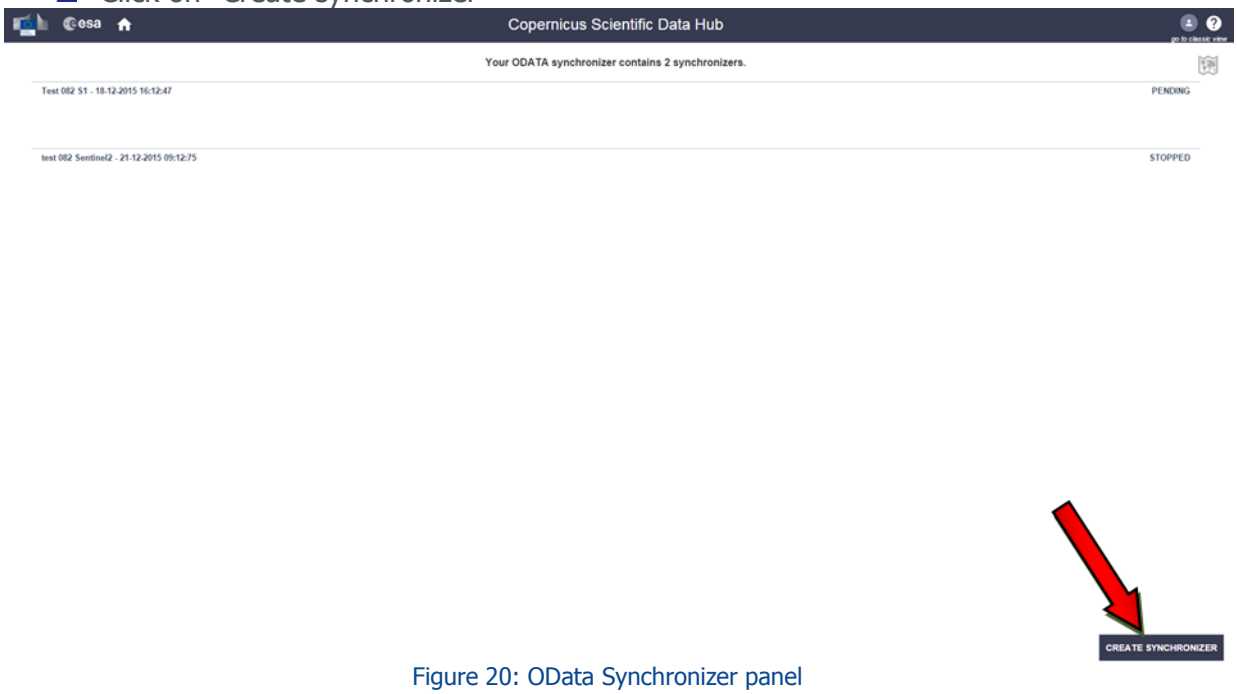

Figure 20: OData Synchronizer panel

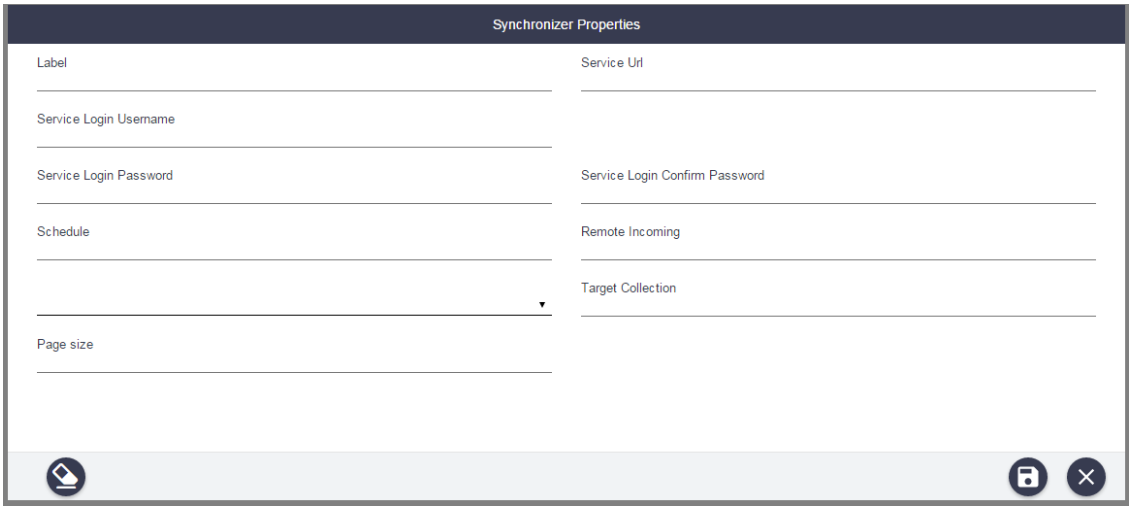

Figure 21: Create an OData Synchronizer

- Fill the records as follows:
  - Label= Name of the synchronizer
  - Service URL= `https://[Back-End_DHuS_address]/odata/v1`
  - Service Login Username= User name of a user registered in the back end which ha the archive manager rights
  - Service Login Password= password of the user in the previous step
  - Schedule= how often the synchronizer shall be running. The syntax of the crons is explained in the `dhus.xml` header (section APPENDIX A)
  - Remote incoming=absolute path of the back end DHuS incoming
  - Request= "start" or "stop"
  - Page size= number of products synchronized at each synchronizer run
- Click on the button with the "floppy disk" shape

How to update a Synchronizer?

The Administrator shall

- Log in as Root in the front end DHuS instance and select the tab "user profile"
- Select the panel "OData synchronizers" and then click on the "pencil" next to the synchronizer to be updated

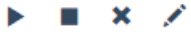

Figure 22: Updating a synchronizer

- Edit the records to be updated
- Click on the button with the "floppy disk" shape

How to delete a Synchronizer?

The Administrator shall:

- Log in as Root in the front end DHuS instance and select the tab "user profile",

- Select the panel "OData synchronizers" and then click on the "X" shaped button next to the synchronizer to be updated.

Next to an existing synchronizer tab, there are also buttons for **starting and stopping** the item. The "play" button is to start the synchronizer; the "square" button is to stop it.

User Manual

The Data Hub Software was fully experienced by user of the Copernicus Data Access instances and ESA provided them online user guides for the Scientific Data Users and for the Collaborative Ground Segment users (see respectively RD 5: and RD 7:).

References

Reference Number	Document name, Reference, issue number, revision number, date
RD 1:	European Earth observation programme Copernicus: http://www.copernicus.eu/
RD 2:	Sentinel Online: https://sentinels.copernicus.eu/web/sentinel/home
RD 3:	Sentinels Scientific Data Hub: https://scihub.copernicus.eu/
RD 4:	Collaborative Data Hub: http://colhub.esa.int
RD 5:	https://scihub.copernicus.eu/userguide
RD 6:	https://colhub.esa.int/userguide/
RD 7:	GitHub open source framework https://github.com/SentinelDataHub/DataHubSystem

Revisions table

Issue	Date	Description	Author
1.0	05-02-2015	First issue of the document	A.G. CASTRIOTTA B. BORGIA
1.1	17-03-2016	The following deletion has been made to be in line with the Software distribution of versions after 0.9.0-2: • GWT GUI deletion • Statistic panel description deletion User manual remanded to the online versions	A.G. CASTRIOTTA

Abbreviations and acronyms

This section controls the definition of all abbreviations and acronyms used within this document. The adopted abbreviations, acronyms and definitions are taken from international standards as ISO, ANSI or ECSS.

AJS	Angular Javascript
API	Application programming interface
BE	Back End
CRUD	Create Read Update Delete persistence operations
ECSS	European cooperation for space standardization
EO	Earth observation
FE	Front End
FTP	File Transfer Protocol
GMES	Global Monitoring for Environment and Security
GWT	Google Web Toolkit
HMA	Heterogeneous Missions Accessibility
PC	Personal Computer
ROI	Region Of Interest
SUM	Software user's manual
SW	Software

Definitions

This section provides the definition of all common terms used within this document. Special attention has been paid to adopt definitions from international standards as ISO, ANSI or ECSS.

Dataset	A dataset is a group of products. Various criteria are used to define a dataset such as : the use of products in a particular domain, the products meta data attributes, the user community.
Footprint	The ground area detected by a sensor.
Product	A product is a file which contains EO data related to a single acquisition.
Project	A unique set of coordinated activities, with definite starting and finishing points, undertaken by an individual or organization to meet specific objectives within defined schedule, cost and performance parameters – ECSS
Query	A precise request for information retrieval with database and information systems.
Software Component	Part of a software product.
Software product	Set of computer programs, procedures, documentation and their associated data.

APPENDIX A CONFIGURATION FILE EXAMPLES

dhus.xml

```
<!DOCTYPE configuration [
  <!ENTITY varFolder "/data/dhus/dhus-software/var">
]>
<configuration xmlns="fr.gael.dhus.database.object.config">
   <!-- Crons configuration. They all have two parameters:
      - active: defines if the cron is currently active or not
      - schedule: defines the schedule of the cron.
         A schedule is configured according to this pattern:
            Seconds Minutes   Hours Day-of-month Month Day-of-week [Year]
         You      can      find      more      information      on      http://www.quartz-
scheduler.org/documentation/quartz-1.x/tutorials/TutorialLesson06
         Here are some simple examples:
            - "0 0 */1 ** ?": every hour.
            - "0 0 9-17 ? * MON-FRI": on the hour nine to five week days.
            - "0 0 0 25 DEC ?": every Christmas Day at midnight
            - "0 0 3 ? * *": every day at 3 AM -->
   <crons>
      <!-- Cron used to synchronize local archive, defined in system/archive/@path -->
      <archiveSynchronization active="false" schedule="0 0 0/5 ? * *"/>
      <!-- Cron used to clean database, like removing old statistics or old not confirmed
users. -->
      <cleanDatabase active="true" schedule="0 0 */1 * * ?">
         <!-- Definition of the time (in days) system is keeping statistics -->
         <statistics keepPeriod="0" />
         <!-- Definition of the time (in days) for user to confirm its registration -->
         <tempUsers keepPeriod="10" />
      </cleanDatabase>
      <!-- Cron used to dump database -->
      <dumpDatabase active="true" schedule="0 0 3 ? * *" />
      <!-- Cron used to clean database dumps.
         - keep: defines how dumps are stored. -->
      <cleanDatabaseDump active="true" schedule="0 0 4 ? * *" keep="10" />
      <!-- Cron used to evict products when it is required -->
      <eviction active="true" schedule="0 */2 * * * ? " />
      <!-- Cron used to execute user saved filescanners.
         - sourceRemove: defines if found products shall be removed from source -->
      <fileScanners active="true" schedule="0 0 */1 * * ?" sourceRemove="false"/>
      <!-- Cron used to execute user saved searches and send results to users -->
      <searches active="false" schedule="0 */1 * * * ?" />
      <!-- Cron used to send system logs.
         - addresses: logs recipients addresses. They shall be coma-separated -->
      <sendLogs active="false" schedule="0 */1 * * * ?" addresses="dhus.help@serco.com" />
      <!-- Cron used to check all system coherence, including database optimization -->
      <systemCheck active="true" schedule="0 30 23 ? * *"/>
   </crons>
   <messaging>
      <!-- Mail configuration. These values are used only at first launch of
         the system. They shall be modified in Management Panel if needed.
         - onUserCreate: defines if system send mail when creating user
         - onUserUpdate: defines if system send mail when updating user
```

```
         - onUserDelete: defines if system send mail when deleting user -->
      <mail onUserCreate="true" onUserUpdate="true" onUserDelete="true">
         <!-- Mail server definition.
            - smtp: server address
            - port: server port
            - tls: defines if server is using TLS protocol
            - username and password : connection information -->
         <server smtp="xxx.yyy.zz" port="25" tls="false" username="dhus.xxxx@xxx.zz"
            password="xxxxxxx">
            <!-- Information used in "from" part of sent mails.
               - name: displayed name of "from" part
               - address: displayed address of "from" part -->
            <from name="xxxx" address="dhus.xxxx@xxx.zz" />
            <!-- Defines the "reply to" address of sent mails -->
            <replyTo>dhus.xxxx@xxx.zz</replyTo>
         </server>
      </mail>
   </messaging>
   <!-- Network configuration -->
   <network>
      <outbound>
         <channel name="PriorityChannel" weight="75">
            <classifier>
               <includes>
                  <include>
                     <userEmailPattern>.*@gael.fr</userEmailPattern>
                  </include>
               </includes>
            </classifier>
         </channel>
         <channel name="SelfRegisteredChannel" weight="25">
            <defaultUserQuotas>
               <!-- maxConcurrent defines the maximum simultaneous accepted
                   transfers. -->
              <maxConcurrent>5</maxConcurrent>
               <!-- maxCount defines the maximum number of accepted transfers on
                   a customizable period.
                   "periodUnit" attribute defines the unit of the period. Possible
                   units are "DAYS", "HOURS", "MICROSECONDS", "MILLISECONDS",
                   "MINUTES", "NANOSECONDS", "SECONDS".
                   "period" attribute is a sliding time window used to count
                   number of transfered product from now to the past period delay. -->
               <maxCount period="5" periodUnit="DAYS">50</maxCount>
               <!-- Maximum accepted size of transfer file. -->
               <maxSize>10737418240</maxSize>  <!-- 10Gb -->
               <!-- maxCumulativeSize defines the maximum cumulated accepted
                   transfers size on a customizable period.
                   period/periodUnit attributes are defined in maxCount description. -->
              <!--                                <maxCumulativeSize               period="5"
periodUnit="MINUTES">1073741824</maxCumulativeSize> -->
               <!-- maxBandwidth maximum bandwidth authorized for this channel. -->
               <maxBandwidth>13107200</maxBandwidth>  <!-- 100Mb/s -->
            </defaultUserQuotas>
         </channel>
      </outbound>
      <inbound>
         <channel name="PriorityChannel" weight="75">
            <classifier>
               <includes>
                  <include>
                     <userEmailPattern>.*@gael.fr</userEmailPattern>
```

```
                  </include>
               </includes>
            </classifier>
         </channel>
         <channel name="SelfRegisteredChannel" weight="25">
            <defaultUserQuotas>
               <!-- maxConcurrent defines the maximum simultaneous accepted
                   transfers. -->
               <maxConcurrent>5</maxConcurrent>
               <!-- maxCount defines the maximum number of accepted transfers on
                   a customizable period.
                   "periodUnit" attribute defines the unit of the period. Possible
                   units are "DAYS", "HOURS", "MICROSECONDS", "MILLISECONDS",
                   "MINUTES", "NANOSECONDS", "SECONDS".
                   "period" attribute is a sliding time window used to count
                   number of transfered product from now to the past period delay. -->
               <maxCount period="1" periodUnit="DAYS">50</maxCount>
               <!-- Maximum accepted size of transfer file. -->
               <maxSize>10737418240</maxSize> <!-- 10Gb -->
               <!-- maxCumulativeSize defines the maximum cumulated accepted
                   transfers size on a customizable period.
                   period/periodUnit attributes are defined in maxCount description. -->
               <!--                       <maxCumulativeSize                      period="7"
periodUnit="DAYS">1099511627776</maxCumulativeSize> -->
               <!-- maxBandwidth maximum bandwidth authorized for this channel. -->
               <maxBandwidth>13107200</maxBandwidth> <!-- 100Mb/s -->
            </defaultUserQuotas>
         </channel>
      </inbound>
   </network>
   <!-- Products configuration
      - publicData: defines if system is launched in public mode or not.
      default is true -->
   <products publicData="true">
      <!-- Download configuration
         - compressionLevel: is a value to build the Deflater, from 1 to 9.
         The compression level defines how rigorously the compressor looks
         to find the longest string possible. As a general rule of thumb:
         - Compressing at the maximum level (9) requires around twice as
         much processor time as compressing at the minimum level (1)
         - For typical input, compressing at the maximum as opposed to
         the minimum level adds around 5% to the compression ratio.
         0 value means no compression.
         - checksumAlgorithms: shall be coma-separated -->
      <download compressionLevel="4" checksumAlgorithms="MD5"/>
      <!-- Quicklook calculation parameters
         - height: height of generated quicklooks
         - width: width of generated quicklooks
         - cutting: allow system to cut image when processing quicklooks -->
      <quicklook height="512" width="512" cutting="false" />
      <!-- Thumbnail calculation parameters
         - height: height of generated thumbnails
         - width: width of generated thumbnails
         - cutting: allow system to cut image when processing thumbnails -->
      <thumbnail height="64" width="64" cutting="false" />
   </products>
   <!-- Search configuration -->
   <search>
      <!-- Geocoder configuration
         - url: geocoder url -->
      <geocoder url="http://nominatim.openstreetmap.org">
```

```
        <!-- Nominatim geocoder configuration
           - boundingBox: defines if the geocoder is querying only
           the bounding box of the matching place from the Nominatim Web
           Service i.e. the four corners encompassing the place.
           Otherwise, it will query the complete polygon boundaries, that
           may have lower performance according the the number of vertices
           composing the place's boundaries.
           - maxPointNumber: maximum number of points that can be returned
           for a polygon -->
        <nominatim boundingBox="false" maxPointNumber="50" />
        <!-- Geoname geocoder configuration
           - username: username used to connect to Geoname -->
        <geoname username="dhus" />
     </geocoder>
     <!-- Odata configuration.
        - maxRows: maximum rows returned by Odata Service -->
     <odata maxRows="50" />
     <!-- Solr configuration.
        path: solr path
        core: solr core name
        schemaPath: solr schema path. Shall be empty.
        synonymPath: path of solr synonyms file -->
     <solr path="&varFolder;/solr" core="dhus" schemaPath=""
        synonymPath="" />
  </search>
  <!-- Server configuration.-->
  <server>
     <!-- External url (protocol://host:port/path) is the url viewed by users
        Used in case of an apache proxy redirection for example
        Empty values mean that server values are those which are viewed by users -->
     <external protocol="" host="" port="" path="" />
     <!-- FTP server configuration.
        port : ftp port
        ftps : using ftps protocol or not -->
   <!--  <ftp port="2121" ftps="false" passivePort="30200-30220" />-->

     <ftp port="2121" ftps="false" />
  </server>
  <!-- System configuration -->
  <system>
     <!-- Definition of principal administrator user.
        If User exists, DHuS will give him all rights at launch, but will not
        his password. This shall be done in Management panel of GUI.
        If User is not existing, DHuS will create it with defined password.
        -->
     <administrator name="root" password="rootpassword" />
     <!-- Definition of local archive path. -->
     <archive path="/data-local">
        <!-- Eviction configuration.
           maxDiskUsage: the maximum disk usage that can be allowed for evictable products
           maxEvictedProducts: the maximum evicted products when running an eviction
           keepPeriod: the minimal time in days -->
        <eviction maxDiskUsage="80" maxEvictedProducts="1000"
           keepPeriod="10" />
        <!-- Definition of incoming folder path and the number of maximum subfolders of
each stage -->
        <incoming path="/PUP3/dhus-test/var/incoming" maxFileNo="10" />
     </archive>
     <!-- Definition of database path and where dumps are stored -->
     <database path="&varFolder;/database/dhus" dumpPath="&varFolder;/database_dump" />
     <!-- Definition of system long name and short name. -->
```

```
      <name long="Data Hub Service" short="DHuS" />
      <!-- Processing configuration.
         corePoolSize: defines maximal number of active threads. Default is 1 -->
      <processing corePoolSize="4" />
      <!-- Definition of support name and mail.
         - registrationMail: used to send the administrative registration information. If
this
         field is not set, DHuS is using support mail.
         These values are used only at first launch of the system.
         They shall be modified in Management Panel if needed. -->
      <support name="DHuS Support Team" mail="dhus@gael.fr" registrationMail="dhus@gael.fr"
/>
      <!-- Definition of tomcat path -->
      <tomcat path="&varFolder;/tomcat" />
      <!-- Executor (background service that executes synchronizers)
          It must be enabled if you want to use the synchronisation feature.
          batchMode: the executor will run the synchronizers until there is no more to
synchronize. -->
      <executor enabled="false" batchModeEnabled="false" />
   </system>
</configuration>
```

Server.xml

```
<?xml version='1.0' encoding='utf-8'?>
<Server port="8005" shutdown="SHUTDOWN">
   <Service name="DHuS-Service">
      <Connector port="8080"
         protocol="org.apache.coyote.http11.Http11NioProtocol"
         maxConnections="1000"
         maxThreads="400"
         keepAliveTimeout="2000"
         URIEncoding="ISO-8859-1"
         compression="on"
         compressionMinSize="1024"

compressableMimeType="application/json,application/javascript,application/xhtml+xml,applica
tion/xml,text/html,text/xml,text/plain,text/javascript,text/css"/>
      <Engine name="DHuS-Engine" defaultHost="localhost">
         <Host name="localhost" appBase="webapps" deployOnStartup="true">
            <Valve className="org.apache.catalina.valves.AccessLogValve"
               prefix="access_log-"
               suffix=".txt"
               directory="logs"
               pattern="%h %l %u %t %r %s %b %I %D"/>
            <!-- Access Filter Settings are
                - pattern: the regular expression to filter user request.
                     i.e. "^.*(/odata/v1/).*$" only manages odata request.
                       or "^((?!/(home|new)/).)*$" consider all request but the UI.
                - useLogger="true|false" show or hide the user access in logger output.
                    This setting does impact keeping internal track of the request.
                - enable="true|false" activate/deactivate the valve.
              -->
            <Valve className="fr.gael.dhus.server.http.valve.AccessValve"
               pattern=".*"
               useLogger="true"
               enable="true"/>
         </Host>
      </Engine>
```

```
   </Service>
</Server>
```

log4j2.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration>
	<Properties>
		<Property                                                                name="pattern"
>[$${sys:fr.gael.dhus.version}][%d{DEFAULT}{UTC}][%-5p]  %m  (%file:%line  -  %t)%n%throwable
</Property>
	</Properties>
	<Appenders>
		<Console name="stdout" target="SYSTEM_OUT">
			<PatternLayout pattern="${pattern}" />
			<Filters>
			    <ThresholdFilter level="DEBUG"/>
			    <ThresholdFilter            level="WARN"            onMatch="DENY"
onMismatch="NEUTRAL"/>
			</Filters>
		</Console>
		<Console name="stderr" target="SYSTEM_ERR">
			<PatternLayout pattern="${pattern}" />
			<Filters>
			    <ThresholdFilter level="WARN"/>
			</Filters>
		</Console>
		<RollingFile name="RollingFile" fileName="dhus.log"
			filePattern="dhus-%d{yyyy-MM-dd}.log">
			<PatternLayout>
				<Pattern>${pattern}</Pattern>
			</PatternLayout>
			<Policies>
				<TimeBasedTriggeringPolicy interval="1" modulate="true" />
			</Policies>
			<Filters>
			    <ThresholdFilter level="DEBUG"/>
			</Filters>
		</RollingFile>
	</Appenders>
	<Loggers>
		<logger name="fr.gael.dhus" level="info"/>
		<logger name="fr.gael.drb.query.FunctionCallExpression" level="debug"/>
		<logger name="org.apache.cxf.jaxrs.utils.JAXRSUtils" level="error"/>
		<logger name="org.apache.solr" level="error"/>
		<Root level="info">
			<AppenderRef ref="stderr" />
			<AppenderRef ref="stdout" />
			<AppenderRef ref="RollingFile" />
		</Root>
	</Loggers>
</Configuration>
```

Start.sh

```
if [ -f start_first.sh ]
then
   /bin/sh start_first.sh
fi

BASENAME=`basename $0`
SCRIPT_DIR=`echo $0| sed "s/$BASENAME//g"`
DHUS_HOME=${DHUS_HOME-`(cd $SCRIPT_DIR; pwd)`}
NATIVE_LIBRARIES=/data/dhus/dhus-software/lib/native/kdu/Linux-x86_64/6.4

java -XX:MaxPermSize=256m -Xms1g -Xmx2g                  \
     -XX:+UseConcMarkSweepGC -XX:+CMSIncrementalMode       \
     -Djava.library.path=${NATIVE_LIBRARIES}               \
     -DArchive.check=false                                 \
     -DArchive.forceReindex=false                          \
     -DArchive.incoming.relocate=false                     \
     -DArchive.incoming.relocate.path=/path/to/relocation  \
     -Duser.timezone=UTC                                   \
     -Dcom.sun.media.jai.disableMediaLib=true              \
     -Dsun.zip.disableMemoryMapping=true                   \
     -Daction.record.inactive=true                         \
     -cp "etc:lib/javax.servlet-api-3.0.1.jar:lib/*" fr.gael.dhus.DHuS &

#     -cp "etc:lib/*" fr.gael.dhus.DHuS &
PID=$!
echo $PID > dhus.pid
wait $PID

if [ $? -eq 8 ]
then
   . $0
fi
```