

# DATIM-OCL Installation and Management

- [Overview](#)
- [A. Installation](#)
  - [A.1. Installing Mediators](#)
    - [A.1.1. Installing Shell Script Mediator](#)
    - [A.1.2. Installing Landing page mediator](#)
    - [A.1.3. Installing IMAP IMPORT mediator](#)
    - [A.1.4. Installing openhim-mediator-ocl mediator](#)
      - [A.1.4.1 Create directories](#)
      - [A.1.4.2 Create config.properties for java app](#)
      - [A.1.4.3 Create certificate](#)
        - [A.1.4.3.a Alternative method to get the certificate \(if the above method doesn't work\)](#)
      - [A.1.4.4. Add Channel, Route, Client using Admin Console](#)
  - [A.2. Installing Mediators - using ansible scripts](#)
    - [A.2.1. Manual steps to config OCL Mediator](#)
      - [A.2.1.1 Get the certificate](#)
      - [A.2.1.2. Add Channel, Route, Client using Admin Console](#)
- [B. Management Operations](#)
  - [B.1. Restart the mediators-](#)
  - [B.2. Check logs](#)
  - [B.3. Updating scripts](#)

## Overview

This document describes the installation procedures for the DATIM-OCL and Results Transformation Service mediators and scripts, including:

Role within DATIM	Mediator	Scripts	API Endpoints
<p>DATIM-OCL Sync</p> <p>Process shell scripts to perform synchronization requests between DATIM DHIS2 and OCL</p>	<p>openhim-mediator-shell-script</p> <p>A generic mediator developed by Jembi that can execute shell scripts.</p>	<ul style="list-style-type: none"> <li>• datim-sync-mer</li> <li>• datim-sync-sims</li> <li>• datim-sync-mechanisms</li> </ul>	
<p>DATIM-OCL Export</p> <p>Used when a person or a computer hits a specific link. It generates exports for IMAP exports and MER landing page on OHIE Metadata Clearinghouse</p>	<p>openhim-mediator-landing-page</p> <p>Extension of the openhim-mediator-shell-script that adds support for including URL parameters in the request that are passed on to the scripts.</p>	<ul style="list-style-type: none"> <li>• datim-imap-export</li> <li>• show-mechanisms</li> <li>• show-mer</li> <li>• show-sims</li> <li>• show-tieredsupport</li> </ul>	
<p>DATIM-OCL IMAP Import</p> <p>Process IMAP (indicator map) import and status requests.</p>	<p>openhim-mediator-imap-import</p>	<ul style="list-style-type: none"> <li>• datim-imap-import</li> <li>• datim-imap-status</li> </ul>	

Results Transformation Service Used by transformation service to map incoming data files from MOH to Datim indicators	openhim-mediator-ocl	??	
--	----------------------	----	--

## A. Installation

### A.1. Installing Mediators

#### A.1.1. Installing Shell Script Mediator

```
sudo git clone https://github.com/jembi/openhim-mediator-shell-script.git /usr/share/openhim-mediator-shell-script
sudo vim /usr/share/openhim-mediator-shell-script/config/default.json # Add openhim username, password and url
cd /usr/share/openhim-mediator-shell-script/
sudo npm install
sudo mkdir /etc/openhim
sudo wget https://raw.githubusercontent.com/jembi/openhim-mediator-shell-script/master/config/default.json
sudo mv default.json /etc/openhim/mediator-shell-script.json
sudo vim /etc/systemd/system/openhim-mediator-shell-script.service
```

```
[Unit]
Description=OpenHIM shell-script mediator
[Service]
User=openhim-core
Group=openhim-core
WorkingDirectory=/usr/share/openhim-mediator-shell-script
ExecStart=/usr/bin/npm start
Restart=always
StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=shell-script-mediator
Environment=NODE_ENV=production
[Install]
```

```
sudo systemctl start openhim-mediator-shell-script
sudo mkdir /opt/openhim-shell-scripts
cd /opt/
sudo git clone https://github.com/OpenConceptLab/ocl_datim.git
sudo chown -R openhim-core:openhim-core /opt/ocl_datim/
sudo chown -R openhim-core:openhim-core /opt/openhim-shell-scripts/
sudo su openhim-core
vim /opt/openhim-shell-scripts/datim-sync-mer.sh
#!/bin/sh
python /opt/ocl_datim/syncmer.py true
vim /opt/openhim-shell-scripts/datim-sync-sims.sh
#!/bin/sh
python /opt/ocl_datim/syncsims.py true
vim /opt/openhim-shell-scripts/datim-sync-mechanisms.sh
#!/bin/sh
python /opt/ocl_datim/syncmechanisms.py true
exit
sudo chmod ug+x datim-sync-mer.sh
sudo chmod ug+x datim-sync-sims.sh
sudo chmod ug+x datim-sync-mechanisms.sh
sudo yum -y install python-pip
cd /opt/ocl_datim/
sudo pip install -r requirements.txt
```

### A.1.2. Installing Landing page mediator

```
sudo git clone https://github.com/maurya/openhim-mediator-landing-page.git /usr/share/openhim-mediator-landing-page
sudo vim /usr/share/openhim-mediator-landing-page/config/default.json #Add openhim username, password and url
cd /usr/share/openhim-mediator-landing-page/
sudo npm install
sudo vim /etc/systemd/system/openhim-mediator-landing-page.service
```

```
[Unit]
Description=OpenHIM landing page mediator
[Service]
User=openhim-core
Group=openhim-core
WorkingDirectory=/usr/share/openhim-mediator-landing-page
ExecStart=/usr/bin/npm start
Restart=always
StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=landing-page-mediator
Environment=NODE_ENV=production
[Install]
```

```
sudo systemctl start openhim-mediator-landing-page
sudo chown -R openhim-core:openhim-core /usr/share/openhim-mediator-landing-page/
sudo chmod ug+x /opt/ocl_datim/show-imap.sh
sudo chmod ug+x /opt/ocl_datim/show-mechanisms.sh
sudo chmod ug+x /opt/ocl_datim/show-merindicators.sh
sudo chmod ug+x /opt/ocl_datim/show-sims.sh
sudo chmod ug+x /opt/ocl_datim/show-tieredsupport.sh
```

### A.1.3. Installing IMAP IMPORT mediator

```
sudo git clone https://github.com/maurya/openhim-mediator-imap-import.git /usr/share/openhim-mediator-imap-import
sudo vim /usr/share/openhim-mediator-imap-import/config/default.json # Add openhim username, password and url
cd /usr/share/openhim-mediator-imap-import/
sudo npm install
sudo vim /etc/systemd/system/openhim-mediator-imap-import.service
```

```
[Unit]
Description=OpenHIM IMAP Import mediator
[Service]
User=openhim-core
Group=openhim-core
WorkingDirectory=/usr/share/openhim-mediator-imap-import
ExecStart=/usr/bin/npm start
Restart=always
StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=imap-import-mediator
Environment=NODE_ENV=production
[Install]
```

Make sure the folder /opt/ocl\_datim/ is owned by openhim-core

```
sudo systemctl start openhim-mediator-imap-import
sudo yum install redis
sudo systemctl enable redis
sudo systemctl start redis
mkdir /opt/ocl_datim/data
sudo vim /etc/systemd/system/celery.service
```

```
[Service]
Type=forking
User=openhim-core
Group=openhim-core
EnvironmentFile=/etc/conf.d/celery
WorkingDirectory=/opt/ocl_datim
ExecStart=/bin/sh -c '${CELERY_BIN} multi start ${CELERYD_NODES} \
  -A ${CELERY_APP} \
  --loglevel=${CELERYD_LOG_LEVEL} ${CELERYD_OPTS}'
ExecStop=/bin/sh -c '${CELERY_BIN} multi stopwait ${CELERYD_NODES}'
ExecReload=/bin/sh -c '${CELERY_BIN} multi restart ${CELERYD_NODES} \
  -A ${CELERY_APP} \
  --loglevel=${CELERYD_LOG_LEVEL} ${CELERYD_OPTS}'
[Install]
WantedBy=multi-user.target
```

```
sudo vim /etc/conf.d/celery
```

```
# Name of nodes to start
# here we have a single node
CELERYD_NODES="worker1, worker2"
# or we could have three nodes:
#CELERYD_NODES="w1 w2 w3"
# Absolute or relative path to the 'celery' command:
CELERY_BIN="/usr/bin/celery"
#CELERY_BIN="/virtualenvs/def/bin/celery"
# App instance to use
# comment out this line if you don't use an app
CELERY_APP="import_manager"
# or fully qualified:
#CELERY_APP="proj.tasks:app"
# How to call manage.py
CELERYD_MULTI="multi"
# Extra command-line arguments to the worker
CELERYD_OPTS="--time-limit=3600 --concurrency=2"
# - %n will be replaced with the first part of the nodename.
# - %I will be replaced with the current child process index
# and is important when using the prefork pool to avoid race conditions.
CELERYD_PID_FILE="/var/run/celery/%n.pid"
CELERYD_LOG_FILE="/var/log/celery/%n%I.log"
CELERYD_LOG_LEVEL="INFO"
```

```
sudo systemctl restart celery
```

#### A.1.4. Installing openhim-mediator-ocl mediator

- Description: Mediator for MOH to PEPFAR conversion with OCL

ssh to test.ohie.datim.org

- Copy the folder `<https://github.com/pepfar-datim/DATIM-OCL/tree/master/src/openhim-mediator-ocl>` to ``~/usr/share/openhim-mediator-ocl``

```
sudo vim /usr/share/openhim-mediator-ocl/config/config.json # Add openhim username, password and url
sudo vim /usr/share/openhim-mediator-ocl/config/mediator.json #Put in proper information under config section
at the bottom of the file.
cd /usr/share/openhim-mediator-ocl/
sudo npm install

#add service and start mediator
sudo vim /etc/systemd/system/openhim-mediator-ocl.service
```

```
[Unit]
Description=OpenHIM OCL mediator
[Service]
User=openhim-core
Group=openhim-core
WorkingDirectory=/usr/share/openhim-mediator-ocl
ExecStart=/usr/bin/npm start
Restart=always
StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=ocl-mediator
Environment=NODE_ENV=production
[Install]
```

```
sudo systemctl start openhim-mediator-ocl
```

Verify the mediator is created: Login to OpenHim Admin Console, click Mediators, you should see openhim-mediator-ocl is started (with green color under Last Heartbeat)

#### A.1.4.1 Create directories

```
sudo mkdir /etc/oclMediator. # stores uploaded import file
sudo mkdir /var/log/upstart
sudo mkdir /var/log/upstart/ocl-java-app. # stores logs for java app
sudo mkdir /usr/local/share/ocl # stores config.properties for java app
sudo mkdir /tmp/MOH_files # stores generated xml /zip files based on import file
#grant access to the folder
sudo chown -R openhim-core:openhim-core /var/log/upstart
sudo chown -R openhim-core:openhim-core /var/log/upstart/ocl-java-app
```

#### A.1.4.2 Create config.properties for java app

```
sudo vim /usr/local/share/ocl/config.properties
```

```
dhis.domain=[https://test.geoalign.datim.org/]
dataStorePath=api/dataStore/MOH_imports/
errorDataStorePath=api/dataStore/MOH_imports_error/
dhis.username=xxxx
dhis.password=xxxx
```

```
pollingTimeout=5
ocl.domain=[https://test.ohie.datim.org:5000]
ocl.token=xxxx
adxPath=/tmp/MOH_files/
archivePath=/var/log/upstart/
datastore.useDataStoreForMappings=false
datastore.useDataStoreForMappings.domain=[https://test.geoalign.datim.org/]
datastore.useDataStoreForMappings.username=xxxx
datastore.useDataStoreForMappings.password=xxxx
maxErrorRowsStored=1000
```

### A.1.4.3 Create certificate

- Login to OpenHim Admin Console from browser on [test.ohie.datim.org](https://test.ohie.datim.org)
- Click Certificates
- + Add Client certificate
- Use `oclServlet` for name
- follow instruction to download 2 files: oclServlet.cert.crt, oclServlet.key.pem
- convert certificate to p12 format using the 2 files downloaded

```
openssl pkcs12 -export -out oclServlet.p12 -inkey oclServlet.key.pem -in oclServlet.cert.crt
```

- Enter a password when prompted. Record the password to be used later in A.1.4.6.1.
- Verify oclServlet is showing under Trusted Certificates in Admin Console
- put the oclServlet.p12 file and the password in test.geoalign (# see A.1.4.6.1)

#### A.1.4.3.a Alternative method to get the certificate (if the above method doesn't work)

- Get the oclServlet.p12 and key (the export password) from test system, such as test.geoalign
- Get the certificate oclServlet.cert.crt file which was used to generate the above .p12 file
- Drop/Upload the certificate file (oclServlet.cert.crt) to test.ohie using OpenHIM AdminConsole -> Certificates
- Restart OpenHIM when prompted
- Put the oclServlet.p12 file and key in test.geoalign (if different)

#### A.1.4.4. Add Channel, Route, Client using Admin Console

- Login to OpenHIM Admin Console on [test.ohie.datim.org](https://test.ohie.datim.org) from browser
- Create oclServlet client on test.ohie
  1. In the test.ohie OpenHIM navigate to the 'Clients'
  2. Click on the '+Client' button
  3. Give the client a name of 'MOH data uploader UI' and ID of 'oclServlet'
  4. Under Client Certificate choose the oclServlet certificate
  5. Check the moh-upload role
    - a. if moh-upload role does not exist, under Add New Role enter "moh-upload"
  6. Click the 'Save Changes' button
- Create MOH data upload channel
  - In test.ohie OpenHIM navigate to the 'Channel' interface
  - Click on the '+Channel' button
  - Give the channel a name of 'MOH data upload channel' and description of 'This channel is used for uploading MOH data on DHIS2 to get PEPFAR mappings'
  - Under Request Matching
    - which URL patterns will match this channel: '/uploadMoh.\*'
    - Which clients should be able to access this channel: 'moh-upload'
    -
  - Under Routes, click Add New Route
    - Route Name: OCL Mediator
    - Route Path: /uploadMoh
    - Host: localhost
    - Port: 3004
    - Click Set Route
  - Click the 'Save Changes' button

#### A.1.4.5. Restart the java app

- Find the processor id at port 8090 for the java app

- `sudo netstat -nlp | grep 8090`
- Kill the processor id
  - *Example:* `sudo Kill 22919`
- restart mediator to start the java app
  - `sudo systemctl start openhim-mediator-ocl`

## A.1.4.6. Configuraton on test.gealign

### A.1.4.6.1. ssh to test.gealign

- upload the certificate oclServlet.p12 file (from 1.4.3. ) to /tmp

```
sudo mkdir /etc/oclServlet
sudo mv /tmp/oclServlet.p12 /etc/oclServlet/oclServlet.p12
```

- create /etc/oclServlet/oclservlet.properties based on <https://github.com/pepfar-datim/DATIM-OCL/blob/master/src/OclServlet/oclServlet.properties>

```
sudo vim :/etc/oclServlet/oclservlet.properties
```

```
remoteSystem.useAsync=1
remoteSystem.domain=https://test.ohie.datim.org:5000
remoteSystem.mohUploadPath=/uploadMoh
remoteSystem.username=[leave empty]
remoteSystem.password=[leave empty]

dhis2.domain=https://test3.global.datim.org
dhis2.username=xxxx
dhis2.password=xxxx

certificate.file=/etc/oclServlet/oclServlet.p12
certificate.key=key/export password from step 1.4.3.
```

### A.1.4.6.2. Login to test.gealign from browser

- Go to System Settings -> Access, add the following to CORS whitelist:
  - <https://test.ohie.datim.org>
  - <https://test.ohie.datim.org:5000>

The MoH Data Import App is ready to use.

### A.1.5. Installing Authentication mediator

- Description: Mediator for JWT token verification and user authentication  
ssh to [test.ohie.datim.org](https://test.ohie.datim.org)
- Copy the folder ``https://github.com/pepfar-datim/DATIM-OCL/tree/master/src/openhim-mediator-authenticator`` to ``usr/share/openhim-mediator-authenticator``.

```
sudo vim /usr/share/openhim-mediator-authenticator/config/config.json # Add openhim username, password and url
sudo vim /usr/share/openhim-mediator-authenticator/config/mediator.json #Put in proper information under config
section at the bottom of the file.
cd /usr/share/openhim-mediator-authenticator/
sudo npm install
```

- add service and start mediator -

```
sudo vim /etc/systemd/system/openhim-mediator-authenticator.service
```

```
[Unit]
Description=OpenHIM Authentication mediator
[Service]
User=openhim-core
Group=openhim-core
WorkingDirectory=/usr/share/openhim-mediator-authenticator
ExecStart=/usr/bin/npm start
Restart=always
StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=authenticator-mediator
Environment=NODE_ENV=production
[Install]
```

```
sudo systemctl start openhim-mediator-authenticator
```

Verify the mediator is created: Login to OpenHIM Admin Console, click Mediators, you should see openhim-mediator-authenticator is started (with green color under Last Heartbeat)

## A.2. Installing Mediators - using ansible scripts

on local machine:

```
sudo git clone https://github.com/pepfar-datim/DATIM-OCL /tmp
sudo vim /tmp/DATIM-OCL/install_scripts/ansible_scripts/ansible_scripts_mediators/mediator_inventory # Change
to proper host names for ohie_server and web_server as needed. #Change ansible_user under Ohio_server and
web_server
sudo vim /tmp/DATIM-OCL/install_scripts/ansible_scripts/ansible_scripts_mediators/variables.yaml
# add github username to ghusername, add all the needed configurations. Set testingBaseFolder if needed for
local or sandbox, leave it empty for test/stage/prod.
cd /tmp/DATIM-OCL/install_scripts/ansible_scripts/ansible_scripts_mediators
#run the script
ansible-playbook --key-file <path_to_ssh_user_github_rsa_pem_file> -i mediator_inventory mediator.yaml # for
<path_to_ssh_user_github_rsa_pem_file> such as ~/.ssh/github_rsa.pem
```

Login to the OpenHIM Admin Console on the Ohie server, Click on Mediators from the left navigation, and verify the following mediators are created and started:

- imap-import
- landing-page
- shell-script
- openhim-mediator-ocl
- openhim-mediator-authenticator

### A.2.1. Manual steps to config OCL Mediator

#### A.2.1.1 Get the certificate

- Get the certificate oclServlet.cert.crt file which was used to generate the /etc/oclServlet/oclServlet.p12 file from gealign.datim.org
- Login to OpenHIM Admin Console at ohie.datim.org, click Certificates from left, drop/Upload the certificate file (oclServlet.cert.crt)
- Restart OpenHIM when prompted

#### A.2.1.2. Add Channel, Route, Client using Admin Console



- Login to OpenHIM Admin Console on ohie.datim.org from browser
- Create oclServlet client on test.ohie
  1. In the test.ohie OpenHIM navigate to the 'Clients'
  2. Click on the '+Client' button
  3. Give the client a name of 'MOH data uploader UI' and ID of 'oclServlet'
  4. Under Client Certificate choose the oclServlet certificate
  5. Check the moh-upload role
    - a. if moh-upload role does not exist, under Add New Role enter "moh-upload"
  6. Click the 'Save Changes' button
- Create MOH data upload channel
  - In test.ohie OpenHIM navigate to the 'Channel' interface
  - Click on the '+Channel' button
  - Give the channel a name of 'MOH data upload channel' and description of 'This channel is used for uploading MOH data on DHIS2 to get PEPFAR mappings'
  - Under Request Matching
    - which URL patterns will match this channel: '/uploadMoh.\*'
    - Which clients should be able to access this channel: 'moh-upload'
    -
  - Under Routes, click Add New Route
    - Route Name: OCL Mediator
    - Route Path: /uploadMoh
    - Host: localhost
    - Port: 3004
    - Click Set Route
  - Click the 'Save Changes' button

#### A.2.1.3 Restart the java app

From terminal connect to the ohie.datim.org server,

- Find the processor id at port 8090 for the java app
  - `sudo netstat -nlp | grep 8090`
- Kill the processor id
  - *Example: sudo Kill 22919*
- restart mediator to start the java app
  - `sudo systemctl start openhim-mediator-ocl`

```
sudo netstat -nlp | grep 8090 #get the processor id
sudo Kill <processor id> #replace with the processor id from above command
sudo systemctl start openhim-mediator-ocl
```

#### A.2.1.4. Login to geoalign from browser

- Go to System Settings -> Access, add the following ohie sites to CORS whitelist:
    - <https://ohie.datim.org>
    - <https://ohie.datim.org:5000>
- The MoH Data Import App is ready to use.

## B. Management Operations

### B.1. Restart the mediators-

### B.2. Check logs

- `tail -fn 1000 openhim-mediator-shell-script.log`
- for OCL Mediator
  - tomcat log location on test.geoalign: /var/log/tomcat
  - java app log on test.ohie: /var/log/upstart/ocl-jva-app

## B.3. Updating scripts

`/etc/conf.d/celery`