



Heliophysics Integrated Observatory

Project No.: 238969

Call: FP7-INFRA-2008-2

Setup of HELIO Components

Definition of Required Capabilities

V0.8

<i>Title:</i>	Setup of HELIO Components
<i>Document No.:</i>	HELIO-UCL-S2-003-RQ
<i>Date:</i>	12 June 2014
<i>Editor:</i>	Kevin Benson , UCL-MSSL
<i>Contributors:</i>	R.D. Bentley (UCL-MSSL)
<i>Distribution:</i>	Project

Revision History

Version	Date	Released by	Detail
Draft	20-Feb-2012	K. Benson	
0.5	17-Sep-2013	K. Benson	Revised to include
0.6	01-Nov-2013	K. Benson	Revised section 1 and 7
0.7	16-May-2014	R.D. Bentley	Changed layout
0.8	12-Jun-2014	K. Benson	Revised

Note: This document will continue to undergo revisions during the implementation phase of HELIO to incorporate the evolution of ideas, changes and improvements.

1	Purpose of Document	1
1.1	Reference Documentation.....	1
2	Initial Installation requirements	2
2.1	Requirements for Components:	2
2.2	Ports:	2
2.3	Running Servlet Container through Apache WebServer via a Proxy:	2
3	Setup of Tomcat	3
4	Virtual Machines	3
5	HQI	4
5.1	HQI Component and Relational Database Service pattern.....	4
5.1.1	<i>HQI Connecting Service</i>	4
5.1.2	<i>Supported Databases</i>	4
5.1.3	<i>Updating Deployed Components</i>	4
5.1.4	<i>Installation of the HQI</i>	5
5.1.5	<i>Setup of the HQI</i>	5
5.1.6	<i>Registering the services</i>	6
5.2	DPAS Service.....	6
5.2.1	<i>Installing the service</i>	6
5.2.2	<i>Registering the services</i>	6
6	Context Service with UWS Interface Connecting Service	7
6.1	Installation of Context Service Component	7
6.2	Setup of Context Service.....	8
6.2.1	<i>Setting up Directories</i>	8
6.2.2	<i>Setting up Internal Storage Directories</i>	8
6.2.3	<i>Changing the Properties</i>	8
6.2.4	<i>Application Description Setup</i>	8
6.3	Testing the CXS	9
6.3.1	<i>Setting Inputs and Outputs</i>	9
6.4	Registration of the CXS	9
7	Registry	10
8	Appendix A – Proxy Forwarding	11
9	Appendix B: Test URLs	12

1 Purpose of Document

This document is designed to provide instructions how to install one or more of the components developed for HELIO. Two particular components, HQI and CXS, are used to connect services that can be installed and immediately bring query results for viewing in the HELIO Front End or other client software. Individual user guides for particular services are being developed as separate documents.

1.1 Reference Documentation

Individual User Guides and Interface specifications could serve as good reference guides and can be found here:

http://www.helio-vo.eu/documents/service_guides_list.php

2 Initial Installation requirements

2.1 Requirements for Components:

- Java (JDK) 1.6 (or greater). Ensure an environment variable JAVA_HOME is pointed to the JDK directory.
- Recommendation is Tomcat. Servlet Container i.e. Glassfish, Tomcat greater than 5.5 version, JBOSS. Most testing has been done on Apache Tomcat 7.0: <http://tomcat.apache.org/> and the setup will conform to this. Any working examples are performed with Tomcat 7.0.
- Virtual Machine will 'not' be a problem. More details on Virtual Machines
- MySQL database setup.
- Heliophysics Event Catalogue (HEC) and Universal Observing Catalogue (UOC) require PostgreSQL. A VM is provided with these two databases setup along with the other http software to connect to them. See Section 5 below.

2.2 Ports:

For HELIO port 80 should be open and accessible, if not allowed at your location, port 8080 is acceptable. Servlet containers can be proxied from an Apache and is a common setup in the HELIO environment. System Administrators often opt for this option as well as it ensures no access to the managing web application of a servlet container.

2.3 Running Servlet Container through Apache WebServer via a Proxy:

Appendix A shows the current example set up used for proxy forwarding at MSSL. Instructions for setting up Proxy forwarding is as follows:

- Edit the config file of the Apache Webserver i.e. /etc/httpd/conf/httpd.conf
- Add a forwarding request at the end of the file (example of an Apache Webserver on the same machine as the tomcat, port 8080 is not open to the public only port 80):
 - ProxyPass /helio-ics/ <http://localhost:8080/helio-ics/>
 - ProxyPassReverse /helio-ics/ <http://localhost:8080/helio-ics/>
- Restart Apache

3 Setup of Tomcat

In order to set up Tomcat 7.0 initial steps are as follows:

- Download Tomcat for your environment:
<http://tomcat.apache.org/download-70.cgi>
- (Optional step), the environment variable CATALINA_HOME should point to the installed directory (base).
- Add a manager-gui role and user, which are defined in the file \$CATALINA_HOME/conf/tomcat-users.xml. This allows access to the manager app on Tomcat located through its base url, i.e. <http://localhost:8080/>. An example of a config file is:

```
<role rolename="manager-gui"/>
<user username="tomcatmgr" password="tomcat" roles="manager-gui"/>
```
- Permissions: Many systems may run Tomcat as a different user, components of HELIO will try to write to a directory. Adding the Tomcat user to a particular group may be required to ensure data is written to the file system.
- If using an Apache webservice to proxy to tomcat then change the \$CATALINA_HOME/conf/server.xml to reference the new proxy. Change the 'Connector' tag to reference the Apache Web Server, this will ensure java code inside tomcat when referencing the context url will get the proper host, i.e.:

```
<Connector port="8080" maxHttpHeaderSize="8192" maxThreads="150"
minSpareThreads="25" maxSpareThreads="75" enableLookups="false"
redirectPort="8443" acceptCount="100" connectionTimeout="20000"
disableUploadTimeout="true" proxyName="msslkz.mssl.ucl.ac.uk"
proxyPort="80" />
```

4 Virtual Machines

Virtual Machines may be used for installations of the HELIO components. If 'root' is not available for the primary administrator of the HELIO components, then a system administrator should ensure a setup whereby the primary 'HELIO' login user can restart certain services as described:

- Set up Tomcat
- Set up Databases: MySQL (and Posgres for HEC/UOC installations)
- Apache – Proxy forwarding is normally setup from an Apache installation 'NOT' on the Virtual Machine. Only Proxy forwarding (See Section 2.3 and Appendix A) is all that is needed for HELIO in the apache config. A contact person should be given to the main HELIO contact for any problems.
- HEC (ONLY) Apache. HEC already comes with a Virtual Machine and you can find a download here: http://msslkz.mssl.ucl.ac.uk/usdownloads/helio/vm_data/ NOTE this VM also has an Apache to run a 'php' gui for the HEC. Proxy ports may need to be setup to reach this apache. HELIO Admin should be able to restart Apache that is located on the VM.

5 HQI

The HELIO Query Interface (HQI) is the primary component in the HELIO Framework because it provides access to the underlying data. The interface follows a particular specification for querying data. Two separate components were developed to conform to the HQI interfaces that allow access to data.

- HQI – Primary component that connects to relational databases.
- DPAS – Component that access external services such as FTP, HTTP directories, and other Web services.

5.1 HQI Component and Relational Database Service pattern

Many of the HELIO services provide access to metadata using a database behind a Web service interface.

If you have a particular relational database that could provide a benefit to the HELIO community follow the instructions below.

The data **MUST** be in a relational database and **MUST** have a compliant JDBC driver.

If you have data that could benefit HELIO, but is **NOT** in a relational database see the section on 'Other ways of connecting data and the DPAS Service'

5.1.1 HQI Connecting Service

The Query Interface is implemented by the HQI Component, which connects to any JDBC compliant relational database. JDBC is a Java Database Connector library for specific relational databases. MySQL and PostgreSQL is the most common database in the HELIO Framework and already bundled as part of the HQI service.

The example below demonstrates how to install the Instrument Location Service – it uses a MySQL database and assumes Tomcat servlet container.

The location: <http://msslkz.mssl.ucl.ac.uk/usdownloads/helio/HelioSetup/> has install and config files ready for helio-ils, helio-ics, and helio-dpas.

5.1.2 Supported Databases

HELIO uses primarily a MySQL database; the exceptions are the HEC and UOC data that use a PostgreSQL database.

MySQL and Postgres databases are setup with two users:

helio_admin	Admin user able to access all the HELIO database capabilities – i.e. insert, update, delete, select.
helio_guest	Guest user with only SELECT permissions on the HELIO databases

Postgres in the HEC also uses the normal 'postgres' user for read only queries and an 'apache' user for querying the database through the HEC web pages.

<http://msslkz.mssl.ucl.ac.uk/usdownloads/helio/data/>

5.1.3 Updating Deployed Components

To update a deployed component follow these instructions:

- Access Tomcat – i.e. <http://localhost:8080> – and click on ‘Manager’; this shows the deployed applications.
- Hit the ‘Undeploy’ link next to the app that is needs updating.
- Download the new war file and follow the same instructions for installation of a service as described in the Installation of the HQI.

5.1.4 Installation of the HQI

Installation of the HQI for connecting to a Database

- Download war files (best to use wget):
 - <http://msslkz.mssl.ucl.ac.uk/usdownloads/helio/HelioSetup/helio-queryservice-server.war>
- Rename the war file to the desired context, i.e. helio-ils.war
- Install one HQI service but configure multiple properties file with this one HQI service installation:
 - Unzip the war file into a particular directory on the machine. Note this directory will have files written to this particular directory and needs to have appropriate permissions set. Unzip by doing ‘jar xvf {warfile}’. For this example the jar contents were unzipped into a directory located at: “/srv/helio/WebApps/helio-queryservice”
 - Download a sample context xml file.
Found here <http://msslkz.mssl.ucl.ac.uk/usdownloads/helio/HelioSetup/helio-ils.xml>
 - Place this xml file in the \$CATALINA_HOME/conf/Catalina/localhost the context path defined in the xml file is now accessible after a restart of tomcat at <http://localhost:8080/{context}>. ***Note some tomcat installations may have this location as /etc/tomcat6/Catalina/localhost'*
 - The xml file **MUST** be changed to have the 'docBase' pointed to the directory of where you extracted the jar file. You will note in the next section (Setup of HQI) that you will also change the one 'Environment' entry to point to the config file.
`<Context path="/helio-ils" docBase="/srv/helio/WebApps/helio-queryservice" debug="0" privileged="true">`
 - Other HQI component can be installed by using an xml file and pointing the 'docBase' to this same location and having the 'path' attribute set to your new HQI install. i.e (helio-ics): `<Context path="/helio-ics" docBase="/srv/helio/WebApps/helio-queryservice" debug="0" privileged="true">`

5.1.5 Setup of the HQI

By default the HQI has a MySQL and PostgreSQL libraries bundled. If neither MySQL nor PostgreSQL are used, then a JDBC library must be found on your database.

Search the web to find the appropriate driver, this driver will have a filename ending with ‘.jar’. This library should be placed in the \$CATALINA_HOME/common/lib and a restart of Tomcat will be needed to pick up this new library. Optionally you can also place it into WEB-INF/lib of the unzipped jar, if you do not have access to the common/lib directories.

The HQI Admin web pages will help you setup a new properties file for configuring your HQI service. The admin pages will need to know JDBC URL, username and password. The admin pages will walk you through adding tables from your database into the properties file. This properties file needs to be saved on your machine. When installing the HQI service it should match what was defined in the xml file.

Restart Tomcat. Edits to the property file at this point will not require restarting the servlet container.

A sample properties file can be found here for the:

<http://msslkz.mssl.ucl.ac.uk/misc/downloads/HeliOLS.txt>

5.1.6 Registering the services

Please contact Kevin Benson <mailto:kmb@mssl.ucl.ac.uk> – for manual registration at this time.

5.2 DPAS Service

Data Provider Access Service (DPAS) uses the HQI interface, however, the DPAS does not connect to a relational database, but has special config files to search other web services. External web services may not contain advanced query mechanisms, such as 'WHERE' clauses that can be performed in the HQI component. DPAS by default ignores 'WHERE' or 'SQLWHERE' clauses.

Many HELIO related sites have already been developed for access in the community, but had many different ways of access and searching – these services may be HTTP, FTP, or SOAP based services. The DPAS was created to give a more standard way of searching these sites. DPAS, though akin to the HQI component and with a similar interface as the HQI component, is coded differently for access to these sites.

5.2.1 Installing the service

Follow the same instructions as in 5.1.4 using:

<http://msslkz.mssl.ucl.ac.uk/usdownloads/helio/HelioSetup/helio-dpas.war>

Use this xml context file:

<http://msslkz.mssl.ucl.ac.uk/usdownloads/helio/HelioSetup/helio-dpas.xml>

**No special configuration file is needed (section 6.2) for the DPAS.*

If you have already a particular site or service that could benefit from connecting with DPAS please contact Kevin Benson <mailto:kmb@mssl.ucl.ac.uk>

5.2.2 Registering the services

Please contact Kevin Benson <mailto:kmb@mssl.ucl.ac.uk> – at this time registration is done manually.

6 Context Service with UWS Interface Connecting Service

The Universal Worker Service interface is used for running command-line applications. Install the context service to run command-line apps. If you feel you have a particular application that the HELIO Community could use then by installing and registering your Context Service other clients will be able to run your application. Do note the CXS cannot run interactive applications; it expects to run an application/script from beginning to end and cannot wait for user input. This installation will step through an application setup on the context called 'cxs'.

6.1 Installation of Context Service Component

Installation of the HQI for connecting to a Database

- Download war files (best to use wget):
 - <http://msslkz.mssl.ucl.ac.uk/misc/downloads/helio-cxs.war>
- Rename the war file to the desired context, i.e. helio-cxs.war
- Install one HQI service but configure multiple properties file with this one HQI service installation:
 - Unzip the war file into a particular directory on the machine. Note this directory will have files written to this particular directory and needs to have appropriate permissions set. Unzip by doing 'jar xvf {warfile}'. For this example the jar contents were unzipped into a directory located at: "/srv/helio/WebApps/helio-cxs"
 - Download a sample context xml file.
Found here <http://msslkz.mssl.ucl.ac.uk/usdownloads/helio/HelioSetup/helio-cxs.xml>
 - Place this xml file in the \$CATALINA_HOME/conf/Catalina/localhost the context path defined in the xml file is now accessible after a restart of tomcat at <http://localhost:8080/{context}>. ***Note some tomcat installations may have this location as '/etc/tomcat6/Catalina/localhost'*
 - The xml file **MUST** be changed to have the 'docBase' pointed to the directory of where you extracted the jar file. You will note in the next section (Setup of HQI) that you will also change the one 'Environment' entry to point to the config file.

```
<Context path="/helio-cxs" docBase="/srv/helio/WebApps/helio-cxs" debug="0" privileged="true">
```
- Download war files (best to use wget):
- HQI for connecting to a Database:
<http://msslkz.mssl.ucl.ac.uk/misc/downloads/helio-cxs.war>
- Rename the war file to the desired context i.e. cxs.war
- Copy the war files into tomcat webapps directory \$CATALINA_HOME/webapps
- This will automatically unpackage and unzip the container.
- At this stage you should now be able to access the HQI and DPAS pagespages i.e.: <http://localhost:8080/helio-cxs/>
- Done but requires setup of HQI. The next section describes in quick detail of how to setup the CXS. You will find various detailed documentation located as part of your installation i.e. <http://msslkz.mssl.ucl.ac.uk/cxs/doc/index.jsp>

6.2 Setup of Context Service

The setup of the Context Service requires several steps setting up directories and xml files. These instructions are given below.

6.2.1 Setting up Directories

The Context service needs to store information during a process of an application along with config information. Choose a location that the CXS can store and run information. For this sample we are choosing:

/data/cxs/config -- This will hold the configuration

/data/cxs/execute – This directory is NOT required, but common to setup wrapping shell scripts here to call applications.

/data/cxs/temp – This directory is where CXS will store output and if needed input.

/data/cxs/records – This holds information of the original Job Description received by the CXS.

6.2.2 Setting up Internal Storage Directories

CXS when ran stores all results in the 'temp' directory setup above. Clients can send results to a VOspace, though HELIO tends to use the internal storage mechanism for retrieving results. We must create a link to the 'temp' directory. You can set this up on an separate HTTP url site such as Apache or follow directions below to use the CXS directory.

- Go to \$CATALINA_HOME/webappas/cxs
- Create a link or directory pointing to the temp directory i.e. jobs. 'ln -s /data/cxs/temp jobs'
- Change the properties as described in 7.2.3, which will set a property to the URL of the internalstorage URL.

6.2.3 Changing the Properties

Properties must be setup to tell the CXS where directories are located and registry information. Also to allow internal storage we need to tell Tomcat to allowLinking to setup a symbolic link for storage. Follow these instructions:

1) Download a sample context file here:

<http://msslkz.mssl.ucl.ac.uk/misc/downloads/cxs.xml>

2) The xml file is self-described and will step-thru your needed instructions.

3) Rename it to be the context name you installed i.e. cxs.xml

4) Copy it into the \$CATALINA_HOME/conf/Catalina/localhost

6.2.4 Application Description Setup

The most complex part of the setup is the app-description.xml, which describes each application. An application can have any number of inputs and outputs and it can have different interfaces, such as simple for only a few basic inputs and outputs or complex adding a variety of detailed input and outputs for a complex user. An app-description.xml file and a registratin-template.xml must be placed in the 'config'

directory i.e. /data/cxs/config. Detailed information of the app-description.xml file can be found here: <http://msslkz.mssl.ucl.ac.uk/cxs/doc/ApplicationConfiguration.jsp>

A sample of a HELIO self-describing xml file is located here:

<http://msslkz.mssl.ucl.ac.uk/misc/downloads/cxs/app-description.xml>

6.3 Testing the CXS

6.3.1 Setting Inputs and Outputs

Testing of the CXS can be performed with two different methods.

1. HELIO has provided a simple testing app to test particular CXS installations. It will require you to have knowledge of your desired inputs, name of the registered application, and the location of the actual CXS service. You can download the application at <http://msslkz.mssl.ucl.ac.uk/helio/cxs/simpletester.zip> This small application requires 'Java' to run on your machine. It is given with a sample calling of a CXS that can be tested.
2. CXS once installed comes with its own testing webpage to call applications internally in the CXS. Go to the location of your service with your web browser and click 'Test run of application' in the menu.

6.4 Registration of the CXS

Please contact Kevin Benson <mailto:kmb@mssl.ucl.ac.uk> – at this time it is easier for a manual registration to occur.

7 Registry

Registry contains metadata on service and other resources to help discover services and the capabilities. The Registry Component follows the IVOA standards and not expected to be installed by other HELIO services. A centralized HELIO publishing registry is already provided. Contact Kevin Benson <mailto:kmb@mssl.ucl.ac.uk> to begin the procedure of registering a service. If registration will undergo many changes then permissions will be given as needed to allow changes a client to make his/her own changes onto the publishing registry.

8 Appendix A – Proxy Forwarding

Below is an example used at MSSL for proxy forwarding:

This is the typical proxy forwarding to a tomcat on the same machine.

```
ProxyPass      /helio-ics/ http://localhost:8080/helio-ics/
ProxyPassReverse /helio-ics/ http://localhost:8080/helio-ics/
ProxyPass      /helio-ils/ http://localhost:8080/helio-ils/
ProxyPassReverse /helio-ils/ http://localhost:8080/helio-ils/
ProxyPass      /helio-dpas/ http://localhost:8080/helio-dpas/
ProxyPassReverse /helio-dpas/ http://localhost:8080/helio-dpas/
ProxyPass      /helio_registry/ http://localhost:8080/helio_registry/
ProxyPassReverse /helio_registry/ http://localhost:8080/helio\_registry/
```

Below is again Proxy forwarding for a HEC VM, the VM setup on the same machine as the primary Apache as seen by the outside. The VM was setup with it's own proxy forwarding of 3080->80(apache on the VM) and 4080->8080(tomcat on the VM). Proxy forwarding on the main apache is then forwarding to these other ports.

```
ProxyPass      /hec/ http://localhost:3080/hec/
ProxyPassReverse /hec/ http://localhost:3080/hec/
ProxyPass      /helio-hec/ http://localhost:4080/helio-hec/
ProxyPassReverse /helio-hec/ http://localhost:4080/helio-hec/
ProxyPass      /helio-uoc/ http://localhost:4080/helio-uoc/
ProxyPassReverse /helio-uoc/ http://localhost:4080/helio-uoc/
ProxyPass      /helio_ils/ http://localhost:4080/helio_ils/
ProxyPassReverse /helio_ils/ http://localhost:4080/helio_ils/
ProxyPass      /helio_ics/ http://localhost:4080/helio_ics/
ProxyPassReverse /helio_ics/ http://localhost:4080/helio_ics/
ProxyPass      /helio_dpas/ http://localhost:4080/helio_dpas/
ProxyPassReverse /helio_dpas/ http://localhost:4080/helio\_dpas/
```

9 Appendix B: Test URLs

Below is a list of URLs that work at MSSL, use these urls and change to your own domain to ensure setup is correct:

<http://msslkz.mssl.ucl.ac.uk/helio-ils/HelioQueryService?STARTTIME=2011-10-05T00:00:00&ENDTIME=2011-10-05T00:00:00&FROM=trajectories>

http://msslkz.mssl.ucl.ac.uk/helio-ics/HelioQueryService?STARTTIME=2003-10-20T20:30:56&ENDTIME=2009-10-20T20:30:56&FROM=instrument_observatory

http://msslkz.mssl.ucl.ac.uk/helio-dpas/HelioQueryServlet?STARTTIME=2009-02-14T20:00:00&ENDTIME=2009-02-15T03:59:59&INSTRUMENT=SOHO_EIT