



*Heliophysics  
Integrated  
Observatory*

**Project No.: 238969**  
**Call: FP7-INFRA-2008-2**

**Instrument Capabilities Service  
Developers Guide**  
*Version 0.1*

<i>Title:</i>	<b>Instruments Capabilities Service – Developers Guide</b>
<i>Document No.:</i>	HELIO_UCL_S2_008_TN_ICS
<i>Date:</i>	16 May 2012
<i>Editor:</i>	<b>Kevin Benson, UCL</b>
<i>Contributors:</i>	
<i>Distribution:</i>	Project



Revision History

<b>Version</b>	<b>Date</b>	<b>Released by</b>	<b>Detail</b>
0.1	16/05/2012	Kevin Benson	Initial Draft

Note: Any notes here.

1. Introduction .....	1
1.1. Suggested Reading .....	1
2. System Requirements .....	1
3. Service Installation .....	1
3.1. Downloading the Service .....	1
3.2. Building the Service .....	2
3.3. Installing the Service .....	2
3.3.1. Setup of Tomcat .....	2
4. <i>Without 'Apache Webservers'</i> .....	2
4.1.1. With an Apache Webserver that has port 80 open .....	2
4.1.2. Running Tomcat through Apache WebServer via a Proxy: .....	2
4.1.3. Installing the Mysql Data .....	3
4.1.4. Installing the HQI Service .....	3
4.1.5. Registration .....	3
5. Database Structure .....	3
5.1. Dependencies .....	3
5.2. Logging .....	4
5.3. Java doc .....	4
5.4. Web Service Tests .....	4
5.5. Other Tests .....	4
6. Maintaining & Extending the Service .....	4
6.1. Fixing problems .....	4
6.2. Adding new Data .....	5
6.3. Download, Build, Install .....	5
6.3.1. Javadocs .....	5
6.3.2. Downloading Source .....	5
6.3.3. Building HQI .....	5
7. Appendix A – Admin Gui .....	6

## 1. Introduction

The Instrument Capabilities Service (ICS) was created as a helper service to enable more precise data requests to the Data Provider Access Service (DPAS). The ICS allows a client to determine which instruments are capable of viewing any particular event. The ICS can specify which instruments are available and therefore able to see a particular event or set of event(s), but does not give instrument location. The ICS was created to further help the client by filtering down the location, with the ICS a client can be certain which instruments should be able to view the event further allowing the client to easily obtain the data ‘if needed’ from the DPAS.

### 1.1. Suggested Reading

<i>To build the service</i>	
Interface Specification	<a href="#">Interface Doc/Service Interface Specification.docx</a>
Helio API	<a href="#">Helio API</a>
Java (compile service)	<a href="http://www.oracle.com/technetwork/java/javase/overview/index.html">http://www.oracle.com/technetwork/java/javase/overview/index.html</a>
Tomcat (web container to host the service)	<a href="http://tomcat.apache.org/">http://tomcat.apache.org/</a>
Maven (build system)	<a href="http://maven.apache.org/">http://maven.apache.org/</a> (or use a plug-in to your development environment)
<i>To extend the service</i>	
SOAP + WSDL (web service definition)	<a href="http://www.w3.org/TR/wsdl">http://www.w3.org/TR/wsdl</a> (or use a plug-in to your development environment)
ICS Database Design	Helio_ICS_ILS.doc

## 2. System Requirements

- Java 1.6
- Servlet Container i.e. Tomcat greater than 5.5, GlassFish, JBOSS. For the purpose of this document, installation is in Tomcat.
- JDBC Compliant Database: If setting up copies of ICS and ILS databases then the dump files would come from a MySQL database.
- Preference of port 80. Second preference is port 8080. Optional use of Apache as a web proxy to a servlet container (Tomcat) is described below.

## 3. Service Installation

ICS primary installation is Mullard Space Science Laboratory (MSSL), instructions will setup a copy/mirror of the ICS.

### 3.1. Downloading the Service

- HQI Service to be installed on a servlet container:

- <http://sourceforge.net/projects/helio-vo/files/helio-queryservice-5.0.war/download>
- Config file for HQI service, primary source MSSL:
  - [http://sourceforge.net/projects/helio-vo/files/ics\\_ils\\_installs.zip/download](http://sourceforge.net/projects/helio-vo/files/ics_ils_installs.zip/download)
  -
- Mysql Dump, primary source MSSL:
  - [http://msslkz.mssl.ucl.ac.uk/downloads/ics\\_ils\\_mysql\\_dump.zip](http://msslkz.mssl.ucl.ac.uk/downloads/ics_ils_mysql_dump.zip)

### **3.2. Building the Service**

No building is necessary.

### **3.3. Installing the Service**

It is assumed installation of a servlet container Tomcat is installed. Extra information about web proxying via Apache is given as an optional benefit.

#### **3.3.1. Setup of Tomcat**

You will need to add a manager-gui role and user defined in the \$CATALINA\_HOME/conf/tomcat-users.xml. This would allow you to access the manager app on tomcat located at the base url of tomcat i.e. <http://localhost:8080/>

An example of a config file is:

```
<role rolename="manager-gui"/>
<user username="tomcatmgr" password="tomcat" roles="manager-gui"/>
```

## **4. Without 'Apache Webserver'**

- Tomcat must have port 8080 open or change the conf/server.xml and have port 80 open (this would require running as root).

#### **4.1.1. With an Apache Webserver that has port 80 open**

- Tomcat can be run on any port and does not require the port to be open, as long as the Apache Webserver is proxying requests. See installation instructions on 'Running Tomcat through Apache'

#### **4.1.2. Running Tomcat through Apache WebServer via a Proxy:**

- 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/>
- Go to tomcat conf directory \$CATALINA\_HOME/conf and edit server.xml
- 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" />
```

- Restart Apache and Tomcat

### 4.1.3. Installing the Mysql Data

This installation assumes a helio-ics database on a mysql server on the current machine and user ONLY has SELECT permissions on the database. This installations assumes a user and password of 'helio'.

The command below must be ran as an admin of mysql:

```
mysql -u root -password=xxxxx helio-ics < ics_data.txt
```

```
mysql -u root -password=xxxxx helio-ics < ics_ils_data.txt
```

### 4.1.4. Installing the HQI Service

The setup requires a directory on your system to hold various data such as config files and votables, i.e. /data/helio. For the rest of the installation instructions this will be referred to as \$HELIO\_BASE.

- Place the helio-hqi.war into \$HELIO\_BASE
- Run 'mkdir \$HELIO\_BASE/helio-ics'
- On a command line run 'cd \$HELIO\_BASE/helio-base' then 'jar xvf ../helio-hqi.war'
- Copy the HQI-ICS-CONFIG.txt into \$HELIO\_BASE
- \* Edit HQI-ICS-CONFIG.txt to point to your mysql assumed on the same machine and to the correct user and passwords. This might be optional if you followed exact directions above.
- Copy the helio-ics.xml into \$CATALINA\_HOME/conf/Catalina/localhost
- Edit the helio-ics.xml
  - change the 'docBase' attribute to point to the directory of \$HELIO\_BASE/helio-ics
  - Change the property/context environment entry, to have it's value pointed to \$HELIO\_BASE/HQI-ICS-CONFIG.txt
- 
- Restart tomcat.

### 4.1.5. Registration

Contact <mailto://helio-services@majordomo.mssl.ucl.ac.uk> to quickly have your service registered in the helio registry. Mail with the URL of your service.

## 5. Database Structure

See reference to [Helio ICS ILSDBStructure.doc](#)

### 5.1. Dependencies

## 5.2. Logging

Logging of the servlet container \$CATALINA\_HOME/logs/Catalina.out.

Logging overview:

- SQL Queries
- Start and End date times
- Votable names saved.

ICS has a common ‘Admin’ interface for also viewing logs. Access the ICS and main page from: <http://{baseurl}:{port}/helio-ics/> Click on ‘Admin Page’ and you can follow a link for accessing the logs.

## 5.3. Java doc

API javadoc to call HQI services including ICS can be found here:

## 5.4. Web Service Tests

Test of a typical REST query, resulting in votable response.

[http://{baseurl}:{port}/helio-ics/HelioQueryService?STARTTIME=1890-10-20T20:30:56&ENDTIME=2009-10-20T20:30:56&FROM=instrument\\_observatory](http://{baseurl}:{port}/helio-ics/HelioQueryService?STARTTIME=1890-10-20T20:30:56&ENDTIME=2009-10-20T20:30:56&FROM=instrument_observatory)

Test of a long running query, resulting of a votable saved on the file system.

[http://{baseurl}:{port}/LongRunningQueryService?STARTTIME=1992-10-18T20:30:56&ENDTIME=2009-10-20T20:30:56&FROM=earth\\_orbit](http://{baseurl}:{port}/LongRunningQueryService?STARTTIME=1992-10-18T20:30:56&ENDTIME=2009-10-20T20:30:56&FROM=earth_orbit)

## 5.5. Other Tests

Java Test Client can be modified to run a few simple tests on the service, this requires an svn checkout of the helio-queryservice (see instructions below). Once checked out, change the ‘eu.heliovo.queryservice.client.axis.QueryWebServiceClient’ edit the file and change the url at the bottom located in the ‘getCall()’ method to your particular service. Currently it is set to localhost:8080/helio-dpas. This should be changed to helio-ics on your server. The running of the java application can be done on the command line, though Eclipse environment is normally thought to be simpler to add all the classpath 3<sup>rd</sup> party libraries.

Command line without 3<sup>rd</sup> party classpath jars defined:

```
‘java -classpath target/helio-queryservice-client-5.0-SNAPSHOT.jar  
eu.heliovo.queryservice.client.axis.QueryWebServiceClient’
```

## 6. Maintaining & Extending the Service

Updating of the ICS requires following the ‘Installing the HQI Service’ above. Updates of the database can also be done by following ‘Installing of Mysql Data’ described above.

### 6.1. Fixing problems

Contact <mailto://helio-services@majordomo.mssl.ucl.ac.uk>

## 6.2. Adding new Data

New data in existing tables will automatically be available. New tables must be added to the HQI-ICS-CONFIG.txt, this can be done using the admin gui located at <http://{{baseurl}}:{{port}}/helio-ics/admin> or edited manually.

## 6.3. Download, Build, Install

If the requirement is to view the source code to make an extension or correct a possible error then view javadocs and follow instructions below for bulding from the source code repository.

### 6.3.1. Javadocs

The javadocs can be found at ?

### 6.3.2. Downloading Source

The ICS source code is part of the HELIO-vo project in sourceforge. The main page is: <http://sourceforge.net/projects/helio-vo/>. You can download the complete source code using svn with:

```
svn co https://helio-vo.svn.sourceforge.net/svnroot/helio-vo helio-vo
```

ILS uses the helio-queryservice for the code base. Change directory to helio-queryservice.

### 6.3.3. Building HQI

Helio QueryService (HQI) is built using Maven. It is recommended to make a

```
mvn clean
```

before building and

```
mvn package
```

in the helio-queryservice directory.

This will build a war file (helio-queryservice-{{version}}.war) in the helio-queryservice-server/target directory.

## 7. Appendix A – Admin Gui

The Helio Query Interface comes with a bundled Admin GUI to help setup a property file to be configured to your relational database. The Property file for the ICS has already been configured for the database, but the Admin GUI you may.

Access the ‘Admin Page’ from the Helio Query Interface. Click on ‘Configuration for connecting database’. See Fig. 2 for a sample of the inputs of a MSSQL ICS configuration:

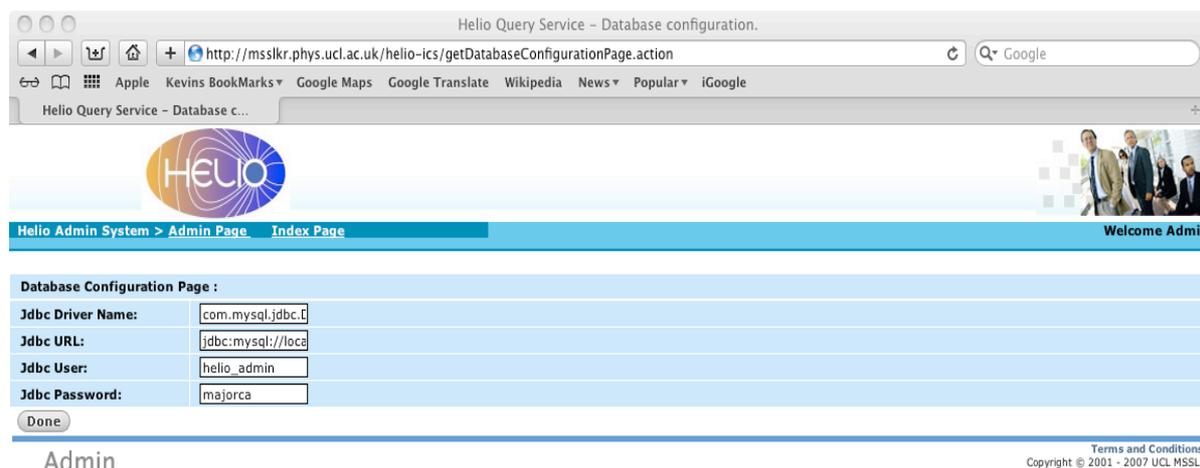


Fig 1. Setting up Database Properties.

Once configured to your database the HQI interface can read all the tables and fields, which would allow the Admin to setup which tables and fields can be viewed or queried. Click on ‘Admin Page’ and choose the link ‘Create Property File’. This page shows a drop down of each table. By choosing a table a user can setup constraints and which columns to be accessed. The ‘Time Constraint’ text box should be filled if your table is based around time.

Once all the tables are configured, choose the location for the property file to be saved.

# Instrument Capabilities Service – Developers Guide

## Version 0.1

Helio Query Service – Database configuration.

http://msslkr.phys.ucl.ac.uk/helio-ics/getConfigurationPropertyFilePage.action

Helio Admin System > Admin Page Index Page

Database tables : instrument\_observatory\_new Done

Jdbc Driver Name: com.mysql.jdbc.Driver

Jdbc URL: jdbc:mysql://localhost

Jdbc User: helio\_admin

Jdbc Password: majorca

Service description: (Description for this service.)

File Name & Path: (Please enter File name and path, where the property file should be saved.)

Added Columns

Table Name	Column Names	Column Description	Column UCD	Column Units	Time Constraint	Instrument Constraint	Coordinate Constraint	Order By Constraint	Max Record Allowed
------------	--------------	--------------------	------------	--------------	-----------------	-----------------------	-----------------------	---------------------	--------------------

List of columns from table

Time Constraint: (Example: start\_date=<=<[:kwstartdate:]> and end\_date=<=<[:kwstartdate:]>)

Instrument Constraint: (Example: ins\_id like [:kwinstrument:] or you can use equal to operator as shown ins\_id=[:kwinstrument:].)

Coordinate Constraint: (Not included now/ Yet to develop.)

Order By Constraint: (Example : order by ins\_id)

Max Record Allowed: (Example : 500)

	Column Name	Column Description	Column UCD	Column Units
<input type="checkbox"/>	name			
<input type="checkbox"/>	observatory_name			
<input type="checkbox"/>	obsinst_key			
<input type="checkbox"/>	experiment_id			
<input type="checkbox"/>	time_start			
<input type="checkbox"/>	time_end			
<input type="checkbox"/>	longname			
<input type="checkbox"/>	inst_type			

Fig 2. Input Properties for ICS table.