



*Heliophysics
Integrated
Observatory*

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

**Context Service
Developers Guide**
Version 0.2

<i>Title:</i>	CXS – Developers Guide
<i>Document No.:</i>	HELIO UCL S2 014 TN CXS
<i>Date:</i>	07 June 2012
<i>Editor:</i>	Kevin Benson , UCL
<i>Contributors:</i>	Bob Bentley, UCL
<i>Distribution:</i>	Project



Revision History

Version	Date	Released by	Detail
0.1	07/06/12	Kevin Benson	Initial Draft
0.2	18/06/14	Bob Bentley	Some tidying

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
3.3.2. <i>Without 'Apache Webservice'</i>	2
3.3.3. With an Apache Webservice that has port 80 open	2
3.3.4. Running Tomcat through Apache WebServer via a Proxy:	2
3.3.5. Installing the CXS Service	3
3.3.6. Registration	3
3.4. Dependencies	3
3.5. Logging	3
3.6. Java doc	3
3.7. Web Service Tests	4
4. Maintaining & Extending the Service	4
4.1. Fixing problems	4
4.2. Adding new Applications	4

1. Introduction

The Context Service (also known as the Universal Worker Service) interface is used for running command-line applications. Install the context service when needing to run simple command-line apps. This installation will step through an application setup on the context called 'cxs'.

1.1. Suggested Reading

<i>To build the service</i>	
Java (compile service)	http://www.oracle.com/technetwork/java/javase/overview/index.html
Tomcat (web container to host the service)	http://tomcat.apache.org/
Maven (build system)	http://maven.apache.org/ (or use a plug-in to your development environment)
<i>To extend the service</i>	
SOAP + WSDL (web service definition)	http://www.w3.org/TR/wsdl (or use a plug-in to your development environment)
Helio Specification	Service Interface Specification.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.
- An Application to run that cannot be interactive.
- 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.
- **The installation procedure below walks through an IDL application.

3. Service Installation

This installation procedure will step you through running a basic IDL application that plots light curves. It is assumed IDL and SolarSoft are installed with the appropriate solar soft environment variables setup.

3.1. Downloading the Service

- CXS Service to be installed on a servlet container:
 - <http://sourceforge.net/projects/helio-vo/files/helio-cxs-5.0.war/download>
- Zip file containing the application and config to be placed in \$HELIO_BASE.
 - Note the actual Application is bundled in an 'execute' directory under \$HELIO_BASE, this is not required. It is only here for convenience for administration of the CXS.
 - http://sourceforge.net/projects/helio-vo/files/cxs_install.zip/download

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"/>
```

3.3.2. 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).

3.3.3. 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'

3.3.4. 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, an example of an Apache Webserver on the same machine as the tomcat, port 8080 (tomcat installation) is not open to the public only port 80 (Apache WebServer):
 - 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

3.3.5. Installing the CXS Service

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

- Place the helio-cxs.war into \$HELIO_BASE
- Run 'mkdir \$HELIO_BASE/helio-cxs'
- On a command line run 'cd \$HELIO_BASE/helio-cxs' then 'jar xvf ../helio-cxs.war'
- Unzip cxs_install.zip and copy all the sub-directories located in 'cxs_base_config' into \$HELIO_BASE. \$HELIO_BASE should now have the additional directories of:
 - **\$HELIO_BASE/config** - This will hold the configuration
 - **\$HELIO_BASE/temp** - This directory is where CXS will store output and if needed input.
 - **\$HELIO_BASE/records** - This holds information of the original Job Description received by the CXS.
 - **\$HELIO_BASE/execute** - This directory is NOT required, but common to setup wrapping shell type scripts here to call applications.
- Edit the \$HELIO_BASE/execute/cron_jobs/do_cxs_goes.sh to modify for any Environment variables that need changing.
- View the \$HELIO_BASE/config/app-description.xml to see how an application is described to the CXS. Change the 'executionPath' to point to your \$HELIO_BASE/execute/cron_jobs/do_cxs_goes.sh
 - This executionPath is critical and points to the application that is ran for the configured application. This is normally noted as a small shell script that wraps around the primary application in previous setups.
- Copy the cxs.xml from the 'tomcat_config' directory located in the zip file into \$CATALINA_HOME/conf/Catalina/localhost
- Edit the cxs.xml
 - change the 'docBase' attribute to point to the directory of \$HELIO_BASE/helio-cxs
 - The xml file is self-explanatory with comments showing what needs to be changed. Certain Environment properties contained in the xml file must be changed to point to \$HELIO_BASE
- Restart tomcat.

3.3.6. 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.

You must edit the \$HELIO_BASE/config/registration-template.xml. This is a self-explanatory to help you register the core metadata about your service.

3.4. Dependencies

3.5. Logging

Logging of the servlet container is written to \$CATALINA_HOME/logs/Catalina.out

3.6. Java doc

<http://www.helio-vo.eu/javadoc/>

3.7. Web Service Tests

Access your CXS service: `http://{baseurl}:{port}/helio-cxs/` On the left hand menu will be a ‘Test Run of Application’ this allows you to add inputs to run the applications. Outputs are preferred to be set to ‘internalstorage://’ which will send back a URL result of outputs stored on the server.

4. Maintaining & Extending the Service

4.1. Fixing problems

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

4.2. Adding new Applications

New applications required editing a special applications xml file called app-descriptions.xml found in \$HELIO_BASE/config directory. The file describes the application and all the input and output parameters along with different interfaces to the application. Different interfaces can be applied to the same application such as a ‘simple’ whereby many parameters use default values or a ‘complex’ interface allowing the user to specify more complex parameters. A self-explanatory app-descriptions-helper.xml is located in the config directory to help build new applications and to be added to the primary app-descriptions.xml.