GOST
Software User Manual

prepared by:	Jorge Fernández-Hernández, Emmanuel Joliet and Nora Garralda
approved by:	Alex Hutton
reference:	GAIA-CU1-UG-ESAC-JFH-005-08
issue:	08
revision:	1
date:	2022-08-23
status:	Issued

Abstract

This document is the Software User Manual for the Gaia Observing Schedule Tool (GOST).

Document History

Issue	Revision	Date	Author	Comment
14	1	2022-08-22	NGT	Changes due to release 22.2: updated information for the script testRestServices.sh, and add new section to describe validation steps.
13	1	2021-09-30	JFH	Changes due to release 22.1: update figures to reflex the new features.
12	1	2020-06-10	JFH	Changes due to release 22.0.1: more information for the scripts launchGostStandAlone.sh and testRestServices.sh.
11	1	2020-05-29	JFH	Changes due to release 22.1: include information for the script CompareGostResults.sh.
10	1	2020-02-10	JFH	Changes due to release 22.0: update property file and figures.
9	2	2020-01-10	JFH	Remove property gaia.cu1.tools.satellite.attitude.rsls.utils.xsd from the property file.
8	1	2019-03-18	JFH	Provide information for the stand-alone service.
7	1	2019-03-18	JFH	Provide information for the new Object Visibility Simple Access service.
6	1	2019-03-18	JFH	Provide information for the new RESTfull API.
5	1	2018-05-18	JFH	Update information for AOCS mode and description of the columns for the output csv.
4	3	2017-02-14	JFH	Additional information for the csv and html output files.
3	2	2016-11-14	JO	Update the documentation for the command line interface (CLI).
2	1	2016-10-28	JO	Include information about the statistic numbers.
1	0	2016-09-27	JFH	Document issued.
D	4	2016-09-27	HS	Minor comments.
D	3	2016-06-27	JFH	Correct minor typos. Add extra information for Aladin Lite.
D	2	2015-06-09	JO	Update urls to new schema
D	1	2014-06-24	EJ	Move installation part to TN EJ-012
D	0	2014-06-09	EJ	First draft

Contents

1 Introduction

1.1 Objectives

This document is the Software User Manual (SUM) for the Gaia Observing Schedule Tool (GOST). The GOST is a tool to provide a means by which the general astronomer can determine the times at which Gaia will observe a given position during the course of its mission.

The tool consists of two parts. A web application permits the user to get the event time of target(s) via a web form interface. The same web application also offers several web services and RESTFull API that can be used to retrieve different metrics, such as the contacts planning data, the number of transits, etc. Also, the Gost provides access to a stand alone service than can directly get accesses through out the distributed jar file.

The web application gets the data from specific GOST data tables. These metrics are populated and updated regularly via the MDB Data Manager (MDB).

1.2 Scope

The scope of this document is to provide generic usage instructions for the GOST. Specific instructions and procedures how to use the tools in the SOC environment in combination with the other software running in the SOC will be covered in section 10.

This issue of the document is applicable for GOST software release 22.0.0.

2 Definitions, acronyms, and abbreviations

The following is a complete list of acronyms used in this document.

The following table has been generated from the on-line Gaia acronym list:

Acronym	**Description**
2MASS	Two-Micron All Sky Survey
AF	Astrometric Field (in Astro)
AOCS	Attitude and Orbit Control Sub-system
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
CCD	Charge-Coupled Device
CSV	Comma-Separated Value (database output format, e.g., for MS Excel)
DEC	Declination
DPAC	Data Processing and Analysis Consortium
EPSL	Ecliptic Pole Scanning Law
ESAC	European Space Astronomy Centre (VilSpa)
FOV	Field of View (also denoted FOV)
FPA	Fractional Pixel Allocation
FoV	Field of View (also denoted FOV)
GB	Gigabit
GOST	Gaia Observation Scheduling Tool
HTML	HyperText Markup Language
ICRS	International Celestial Reference System
IDT	Initial Data Treatment (Image Dissector Tube in Hipparcos scope)
ISO	International Organisation for Standardisation (Geneva, Switzerland)
IVOA	International Virtual-Observatory Alliance
JSP	Java Server Page
MAINDB	MAIN DataBase
MDB	Main DataBase
MJD	Modified Julian Date (to be avoided; see also JD)
MSL	Modified Scanning Law
RA	Right Ascension
REST	REpresentational State Transfer
RSLS	Reference Scanning Law Schedule
SDSS	Sloan Digital Sky Survey
SOC	Science Operations Centre
SRS	Software Requirements Specification

SUM	Spacecraft (or Satellite) User Manual
TCB	Barycentric Coordinate Time
TN	Technical Note
UB	University of Barcelona (Spain)
UI	User Interface
UTC	Coordinated Universal Time
VO	Virtual Object
XML	eXtensible Markup Language

3 Overview

GOST web interface deliver data stored in GOST tables. Those tables are updated externally via the MDM, MDB Data Manager.

A main class will be listening to MDM queue while GOST is registered to it. Whenever a new source or a cross-matched source is being confirmed and ingested in MAINDB, a message is sent. GOST will update metrics table with this information.

See section 10 for more details about configuration properties needed and on how to run the MDM listener.

4 The Web services

GOST web application offers several web services that delivers basic metrics in different output format.

This services are typically servlets running in the application server. They can either be called from browser or CLI with *wget* for instance.

They are accessed with the same url root:

```
https://gaia.esac.esa.int/gost/
```

The available services are described below.

4.1 Gaia Live position in GE

It is mainly designed for outreach. It is located at `https://gaia.esac.esa.int/gost/gkml`

and it accesses the extra query parameter

`https://gaia.esac.esa.int/gost/gkml?time=xxxx`

where the parameter *time* can take the following formats:

- Julian year number: Jyyyy[.xxxx] [TCB—TG];
- ISO 8601 format: yyyy-mm-ddThh:mm:ss[.ssss] [TCB—TG];
- Julian day number: JDdddddddd[.fffff] [TCB—TG];
- ns since ref epoch: i[iiiii...] [TCB—TG].

The output is rendered as a KML file[1]. For example:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<kml xmlns="http://www.opengis.net/kml/2.2" xmlns:gx="http://www.google.com/
    kml/ext/2.2" xmlns:atom="http://www.w3.org/2005/Atom" xmlns:xal="
    urn:oasis:names:tc:ciq:xsdschema:xAL:2.0" hint="target=sky">
     <Document>
         <name>ESA Gaia satellite</name>
         <open>1</open>
         <styleUrl>#Gaia</styleUrl>
         <Style>
             <BalloonStyle>
                 <text>&lt;text&gt;&lt;center&gt;&lt;b&gt;Gaia&lt;/b&gt;&lt;/
                     center&gt;&lt;br/&gt;Position the 2016-06-27T14:29:56 (
                     UTC)&lt;/text&gt;</text>
             </BalloonStyle>
         </Style>
         <Placemark>
             <name>Gaia</name>
             <description>Gaia position seen from Earth at 1622975.4497314037
                 kms from us the 2016-06-27T14:29:56 (UTC)</description>
             <LookAt>
                 <longitude>5.621809218570988</longitude>
                 <latitude>30.027379116517448</latitude>
                 <altitude>0.0</altitude>
                 <heading>0.0</heading>
                 <tilt>0.0</tilt>
                 <range>10000.0</range>
```

[1]KML is an open standard officially named the OpenGIS KML Encoding Standard (OGC KML). It is maintained by the Open Geospatial Consortium, Inc. (OGC). The complete specification for OGC KML can be found at `http://www.opengeospatial.org/standards/kml/`.

```
            </LookAt>
            <Point>
                <coordinates>5.621809218570988,30.027379116517448</
                    coordinates>
            </Point>
        </Placemark>
    </Document>
</kml>
```

4.2 Gaia contact planning

It delivers the contact planning. It is mainly used by UB in mobile app. It is located at

`https://gaia.esac.esa.int/gost/plnview`

and it also accessible though the botton ”Get Downlink/Contact status (JSON)”.

It accesses the extra query parameter

`https://gaia.esac.esa.int/gost/gkml?time=xxxx`

and

`https://gaia.esac.esa.int/gost/gkml?ics`.

The the parameter *time* can take the formats previously displayed. The parameter *ics* returns the result as a calendar file saved in a universal calendar format.

The output is rendered as a JSON format, self explained structure key-value pair. For example:

```
{ "Next": {
    "Station":"Malargue",
    "End":"2014-06-10T08:00:58 (UTC)",
    "Status":"Downloading to Malargue (MLG)",
    "Start":"2014-06-09T23:21:43 (UTC)",
    "Duration":"8.65h"
  },
  "Contact" : {
    "Status":"Scanning",
    "Request":"2014-06-09T15:00:52 (UTC)"
  }
}
```

4.3 Direct query by name

A direct REST (representational state transfer) query can either be called from CLI or browser skipping the web UI. In order to query for a given object, we should make of the following query parameter:

```
https://gaia.esac.esa.int/gost/GostServlet?name=target
```

where *target* is the object we are interested in. The complete output result is rendered in XML format. For example, for the target m31:

```
https://gaia.esac.esa.int/gost/GostServlet?name=m31
```

In this particular case, it makes used of VizieR name resolver by default. To change it, add a service code:

```
https://gaia.esac.esa.int/gost/GostServlet?name=target&service=code
```

where ”&service=code” is an integer code such

- code=0 corresponds to Vizier[2],
- code=2 corresponds to NED[3],
- code=1 corresponds to Simbad[4].

For example, the output of the parameter query

```
https://gaia.esac.esa.int/gost/GostServlet?name=m31&service=2
```

is rendered as

```
<?xml version="1.0" encoding="UTF-8"?><gost>
  <creationTime>Mon Jun 09 16:16:03 UTC 2014(id=1402330563[ns])</
      creationTime>
  <attitudeDataServerId>OPS_RSLS_0021792 - with segments: EPSL_P [2013-12-01
      T12:00:00 (TCB)], ( /olddisk/jworkspace/Gost/./conf/
      OPS_RSLS_0021792_rsls_launch_minus_5_weeks_epsl_comm_leading_TUNED2014
      -03-12.xml )Segment #0:       new EPSL from 2013-12-01T12:00:00
      .000000000 (TCB), at which time nu = 3.9252311467094363E-16 rad
</attitudeDataServerId>
```

[2]The VizieR service at CDS.
[3]The NASA/IPAC ExtraGalactic Database at the California Institute of Technology (CalTech).
[4](The SIMBAD Astronomical Database at the Centre de Donnes astronomiques de Strasbourg (CDS).

```
<startTime>2013-12-19T09:55:44.848 (TCB)</startTime>
<endTime>2019-06-20T06:13:47.564 (TCB)</endTime>
<targets>
  <target>
    <name>m31</name>
    <id>0</id>
    <coords>
      <ra>0.18648476538416495</ra>
      <dec>0.7202809762030913</dec>
    </coords>
    <startTime>2013-12-19T09:55:44.848 (TCB)</startTime>
    <endTime>2019-06-20T06:13:47.564 (TCB)</endTime>
    <events>
      <event>
        <eventDate>2014-06-03T07:42:25.721 (TCB)</eventDate>
        <details>
          <kind>SM</kind>
          <ccdRow>1</ccdRow>
          <ccdStripNo>2</ccdStripNo>
          <angleToZaxis>0.005535483348610173</angleToZaxis>
          <acCcdCoordinate>154.56390671186844</acCcdCoordinate>
          <fov>FOV2</fov>
          <normAcTransitPosition>0.9182601037550459</normAcTransitPosition
              >
          <scanAngle>2.658288251248049</scanAngle>
          <pfovdir>62.2652275989192 -55.0964882700654</pfovdir>
          <ffovdir>9.925852565866435 41.66440824113985</ffovdir>
        </details>
      </event>
      <event>
        <eventDate>2014-06-03T07:42:31.559 (TCB)</eventDate>
        <details>
          <kind>AF</kind>
          <ccdRow>1</ccdRow>
          <ccdStripNo>3</ccdStripNo>
          <angleToZaxis>0.0055345397379475345</angleToZaxis>
          <acCcdCoordinate>153.46302760545709</acCcdCoordinate>
          <fov>FOV2</fov>
          <normAcTransitPosition>0.9181035730865958</normAcTransitPosition
              >
          <scanAngle>2.6582876302360705</scanAngle>
          <pfovdir>62.37027428803965 -55.17297037326464</pfovdir>
          <ffovdir>9.98732769562838 41.578713606483895</ffovdir>
        </details>
      </event>
      <event>
        <eventDate>2014-06-03T07:42:36.414 (TCB)</eventDate>
        <details>
          <kind>AF</kind>
          <ccdRow>1</ccdRow>
          <ccdStripNo>4</ccdStripNo>
```

```
            <angleToZaxis>0.005533754984318633</angleToZaxis>
            <acCcdCoordinate>152.54748170507182</acCcdCoordinate>
            <fov>FOV2</fov>
            <normAcTransitPosition>0.9179733943857807</normAcTransitPosition
               >
            <scanAngle>2.6582871137715833</scanAngle>
            <pfovdir>62.45794346393168 -55.236507793788526</pfovdir>
            <ffovdir>10.038329433080232 41.50742084030539</ffovdir>
          </details>
        </event>
        ...
        <event>
          <eventDate>2019-06-04T10:48:49.931 (TCB)</eventDate>
          <details>
            <kind>AF</kind>
            <ccdRow>7</ccdRow>
            <ccdStripNo>11</ccdStripNo>
            <angleToZaxis>-0.006304476884000062</angleToZaxis>
            <acCcdCoordinate>941.2769686665949</acCcdCoordinate>
            <fov>FOV1</fov>
            <normAcTransitPosition>-1.0458093599016887</
               normAcTransitPosition>
            <scanAngle>2.650496278163309</scanAngle>
            <pfovdir>11.189611786167962 41.32503010325796</pfovdir>
            <ffovdir>221.23189862531535 26.70356250924403</ffovdir>
          </details>
        </event>
      </events>
      <sm2>39</sm2>
      <sm1>36</sm1>
    </target>
  </targets>
  <message>COMPLETED - Target(s) 1 added (SM1+SM2 75 hits).</message>
  <status>COMPLETED</status>
  <sm2>39</sm2>
  <sm1>36</sm1>
  <ephemId>SOLUTIONID = 1580842634044243973</ephemId>
</gost>
```

See Appendix A for more detail about the element definition of the XML output.

The provided RESTfull API gives the change to get access to the Gost programmatically, for example making use of the *jersey*[5] client in Java.

[5] https://jersey.github.io/

4.4 Direct query by coordinates

The source target can also be queried by its astronomical coordinates ra and dec coordinates in degrees:

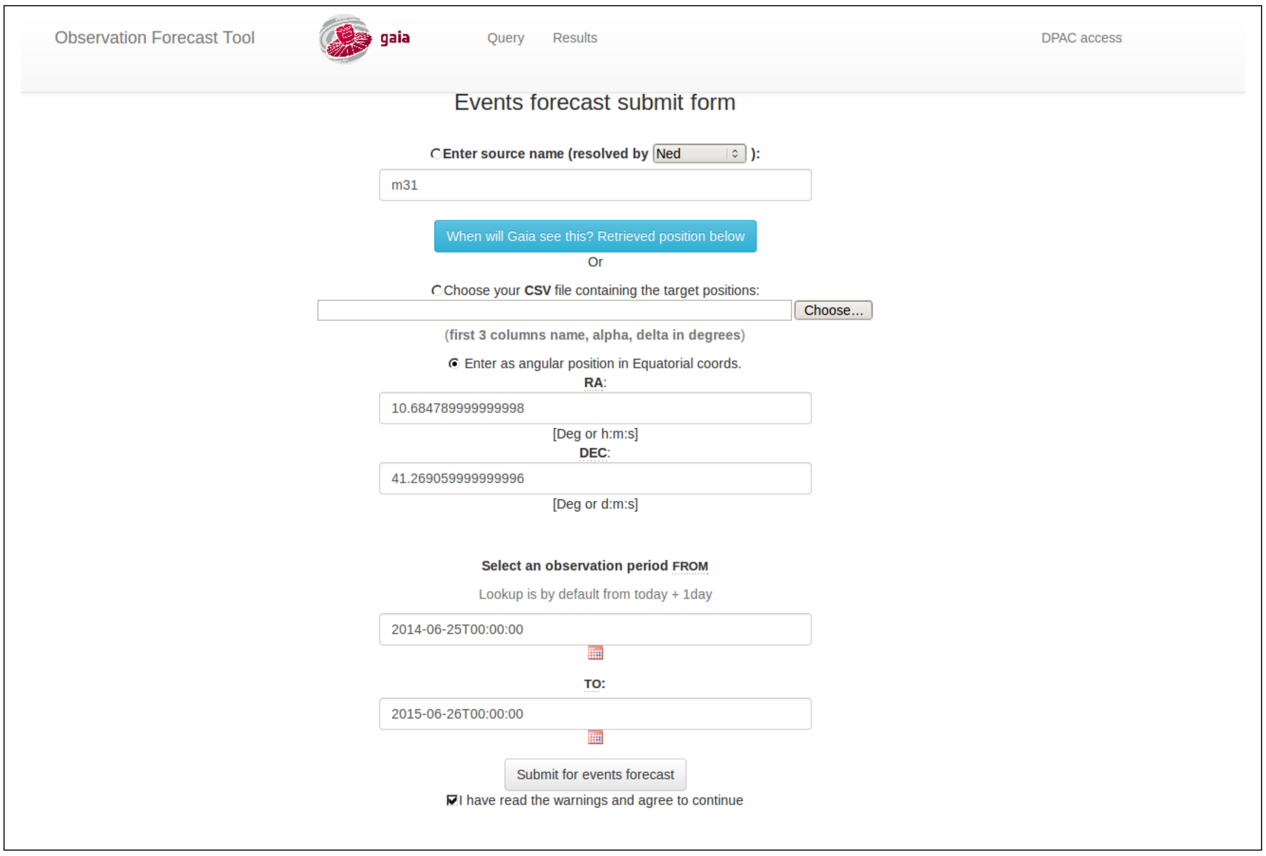
```
https://gaia.esac.esa.int/gost/GostServlet?ra=ra_in_degree&dec=dec_
in_degree
```

The output is rendered in XML format.

See Appendix A for more detail about the element definition of the XML output.

It is also possible to resolve the name of the target making use of the following query

https://gaia.esac.esa.int/gost/GostServlet?name=target&service=code&resolve"



Events forecast submit form

( ) Enter source name (resolved by Ned):
m31
When will Gaia see this? Retrieved position below
Or
( ) Choose your **CSV** file containing the target positions:
Choose...
(first 3 columns name, alpha, delta in degrees)
(●) Enter as angular position in Equatorial coords.
RA:
10.684789999999998
[Deg or h:m:s]
DEC:
41.269059999999996
[Deg or d:m:s]

Select an observation period FROM
Lookup is by default from today + 1day
2014-06-25T00:00:00
TO:
2015-06-26T00:00:00
Submit for events forecast
[✓] I have read the warnings and agree to continue

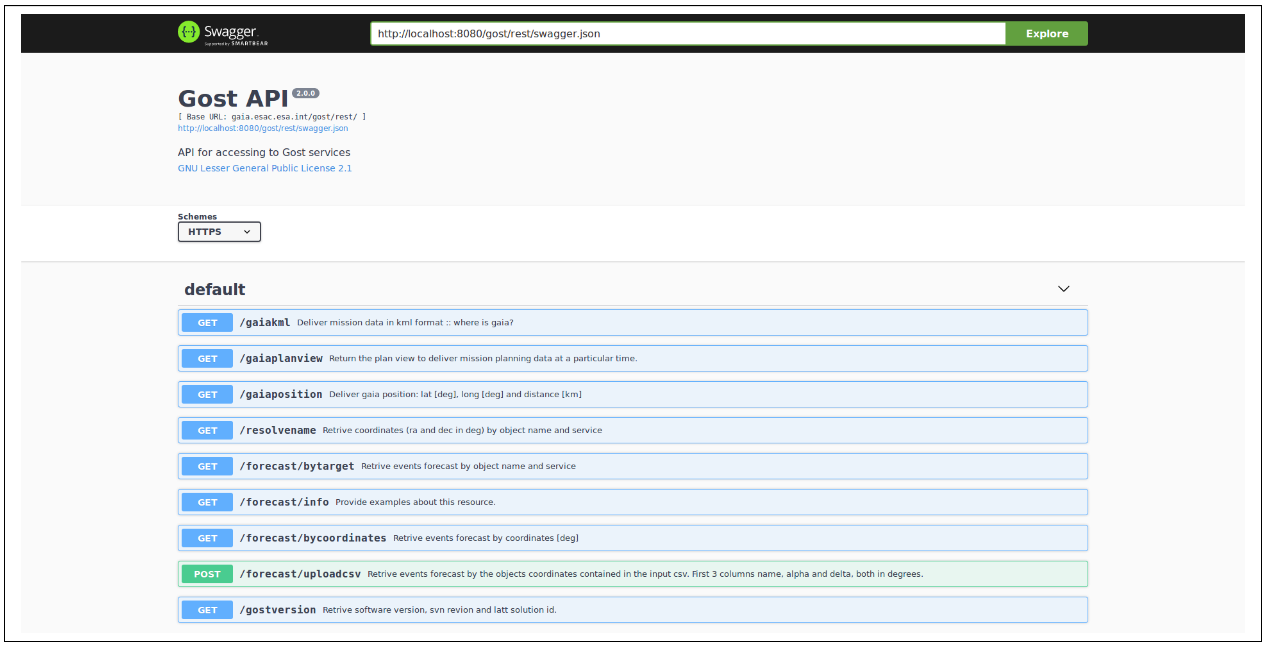
FIGURE 1: Form view page for one year prediction of m31 object.

5 The RESTFull API

The Gost also provides similar services based on a Representational State Transfer (REST) architecture. Note that the access to this service could be restricted to connections from ESAC. The documentation for the provided services can be found in the following link

`https://gaia.esac.esa.int/gost//swagger-ui/dist/index.html`

or through out the button *RESTfull API* (see Figure 5). This button redirects the user to a web-page that contains extensive documentation for the execution of the different provided services (see Figure 2).



http://localhost:8080/gost/rest/swagger.json

Gost API 2.0.0
[Base URL: gaia.esac.esa.int/gost/rest/]
http://localhost:8080/gost/rest/swagger.json

API for accessing to Gost services
GNU Lesser General Public License 2.1

Schemes: HTTPS

default

GET	`/gaiakml`	Deliver mission data in kml format :: where is gaia?
GET	`/gaiaplanview`	Return the plan view to deliver mission planning data at a particular time.
GET	`/gaiaposition`	Deliver gaia position: lat [deg], long [deg] and distance [km]
GET	`/resolvename`	Retrive coordinates (ra and dec in deg) by object name and service
GET	`/forecast/bytarget`	Retrive events forecast by object name and service
GET	`/forecast/info`	Provide examples about this resource.
GET	`/forecast/bycoordinates`	Retrive events forecast by coordinates [deg]
POST	`/forecast/uploadcsv`	Retrive events forecast by the objects coordinates contained in the input csv. First 3 columns name, alpha and delta, both in degrees.
GET	`/gostversion`	Retrive software version, svn revion and latt solution id.

FIGURE 2: Documentation for the RESTfull API.

6 The Object Visibility Simple Access service

The Object Visibility Simple Access Protocol (ObjVisSAP) is an IVOA Data Access protocol which defines the standard for retrieving object constraint-free visibility time intervals through a uniform interface within the VO framework for given object coordinates to be observed by Gaia[6]. In particular the Gost provides this service based on the version 0.4 of the protocol.

The service is provided through the following web services:

[6]See the standard *ObjVisSAP* in `https://wiki.ivoa.net/twiki/bin/view/IVOA/IvoaDAL`.

1. https://gaia.esac.esa.int/gost/ObjVisSAP/examples
2. https://gaia.esac.esa.int/gost/ObjVisSAP/capabilities
3. https://gaia.esac.esa.int/gost/ObjVisSAP/availability
4. https://gaia.esac.esa.int/gost/ObjVisSAP/gaiaobjvisap

The first three services provide meta-data on the VO service as described in the VO document (see ref in footnote[6]). The last service is a synchronous web service that accepts the following parameters for a request that may be submitted using an HTTP GET (query string) or POST action. The default output format is a VOTable. Other output formats can be specified by the RESPONSEFORMAT parameter. In particular for the GET action the following parameters, as defined by the VO standard version 0.4, are accepted (please note: the parameter names are case sensitive):

1. s_ra, the right ascension in degrees contained in the range [0, 360];
2. s_dec, the declination in degrees contained in the range [-90, 90];
3. t_min, the start time to check for object visibility in MJD (modified Julian date);
4. t_max, the end time to check for object visibility in MJD;
5. MAXREC, allows the client to limit the number or records in the response. For the special case of MAXREC=0, the service respond with metadata-only;
6. RESPONSEFORMAT, that allows the client to select the output formats (see Table 2 for more details.

Note that the RESPONSEFORMAT parameter is not compulsory, and the client can also select the output format by the setting the “content-type” in the header (see examples below).

The service can also be accessed by a POST action:

1. by the definition of a csv file, plus an optional value for the RESPONSEFORMAT parameter;
2. by a *form* that contains the same information of the GET action.

The csv file has to have a header with the values in the following order

```
s_ra, s_dec, maxRec, t_min, t_max, min_vis,max_vis, elevation, moon_sep
```

only the first five columns are read. Only two first columns can not be empty. All the rows have to have the same values for the the third, fourth and fifth columns. This is done in order to keep the same logic with the GET action.

TABLE 2: Allowed values for the RESPONSEFORMAT parameter.

media type	short form
application/x-votable+xml;charset=utf-8;serialization=BINARY	votable/b1
application/x-votable+xml;charset=utf-8;serialization=BINARY2	votable/b2
application/x-votable+xml;charset=utf-8;serialization=FITS	votable/fits
application/x-votable+xml;charset=utf-8;serialization=TABLE	votable/td
text/xml;charset=utf-8	votable/xml

6.1 Examples

In this section we provide a number of examples for Object Visibility Simple Access service (making use of the command `curl`).

1. For the source m31, get a votable with the body compressed in different ways setting the accept field in the header:

   (a) ascii,

   curl -X GET ”https://gaia.esac.esa.int/gost/ObjVisSAP/gaiaobjvisap?s_ra=10.68470833&s_dec=41.26875” -H ”accept: application/x-votable+xml;charset=utf-8;serialization=TABLE”

   (b) fits,

   curl -X GET ”https://gaia.esac.esa.int/gost/ObjVisSAP/gaiaobjvisap?s_ra=10.68470833&s_dec=41.26875” -H ”accept: application/x-votable+xml;charset=utf-8;serialization=FITS”

   (c) binary-2 format,

   curl -X GET ”https://gaia.esac.esa.int/gost/ObjVisSAP/gaiaobjvisap?s_ra=10.68470833&s_dec=41.26875” -H ”accept: application/x-votable+xml;charset=utf-8;serialization=BINARY2”

2. Print out only the metadata making use of MAXREC=0

curl -X GET ”https://gaia.esac.esa.int/gost/ObjVisSAP/gaiaobjvisap?MAXREC=0&s_ra=10.68470833&s_dec=41.26875”

3. Filter the number of returned observations making use of MAXREC (=2, for example)

curl -X GET ”https://gaia.esac.esa.int/gost/ObjVisSAP/gaiaobjvisap?MAXREC=2&s_ra=10.68470833&s_dec=41.26875” -H ”accept: application/x-votable+xml;charset=utf-8;serialization=TABLE”

4. Filter by time making use of `t_min` and `t_max`

curl -X GET ”https://gaia.esac.esa.int/gost/ObjVisSAP/gaiaobjvisap?s_ra=10.68470833&s_dec=41.26875 &t_min=57388.0&t_max=57540.0” -H ”accept: application/x-votable+xml;charset=utf-8;serialization=TABLE”

5. Search for visibilities for multiple sources (for example m31 and m50):

curl -X GET ”https://gaia.esac.esa.int/gost/ObjVisSAP/gaiaobjvisap?s_ra=10.68470833,105.697916670&s_dec=41.26875,-8.337777780” -H ”accept: application/x-votable+xml;charset=utf-8;serialization=TABLE”

6. Making use of POST with a csv file `inputFile.csv`

curl -X POST ”https://gaia.esac.esa.int/gost/ObjVisSAP/gaiaobjvisap” -H ”accept: application/x-votable+xml;charset=utf-8; serialization=TABLE” -H ”Content-Type: multipart/form-data” -F ”file=@inputFile.csv ;type=application/x-webarchive”

7. Making use of POST and a form, where only the coordinates of the source m31 are used

curl -X POST ”https://gaia.esac.esa.int/gost/ObjVisSAP/gaiaobjvisap” -H ”accept: application/x-votable+xml;charset=utf-8; serialization=TABLE” -H ”Content-Type: application/x-www-form-urlencoded” -d ”s_ra=10.68470833&s_dec=41.26875”

7 The Web interface

The web interface consist mainly in a web-form accessed via the main page `index.jsp`[7]. The user input should be:

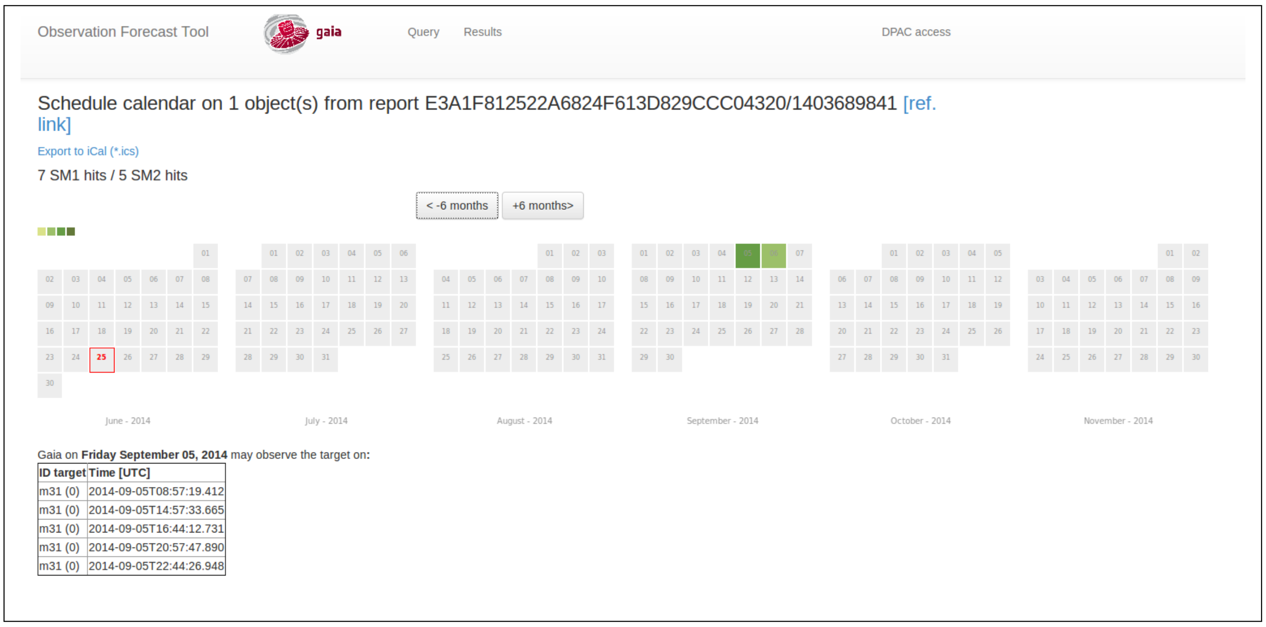
- the target name,
- the astronomical coordinates (ra and dec) in degrees,
- a 3 column CSV text file, each line with an identifier, right ascension and declination.

For example figure 1 displays the web-form for the target m31 (resolved by clicking on the blue button below target name) and period of one year prediction.

The results will appear in the `result.jsp` page (see figure 3). This page keeps an history of the previous query result. The table displays the report id, status, message, created date, coverage period prediction time and links to other output format (XML, CSV,etc.). See Appendix A for more details. When clicking on the id number of the report on the most left-hand column ’ID’ cell, the calendar view will open in a different window showing the result in a calendar output

[7]JSP is a file extension for Java Server Pages file format. A JSP is an HTML page containing a reference to Java servlets, or, java server side applets.

format. This is the main report view. Each cell dark coloured represent a potential observation. A detail result of the events can be displayed by clicking on a particular darker cell (day of predicted observation). The calendar can be navigated by two buttons above on 6 months basis. Example of the calendar view for m31:



Schedule calendar on 1 object(s) from report E3A1F812522A6824F613D829CCC04320/1403689841 [ref. link]

Export to iCal (*.ics)

7 SM1 hits / 5 SM2 hits

Gaia on **Friday September 05, 2014** may observe the target on:

ID target	Time [UTC]
m31 (0)	2014-09-05T08:57:19.412
m31 (0)	2014-09-05T14:57:33.665
m31 (0)	2014-09-05T16:44:12.731
m31 (0)	2014-09-05T20:57:47.890
m31 (0)	2014-09-05T22:44:26.948

FIGURE 3: Calendar view page for one year prediction of m31 object.

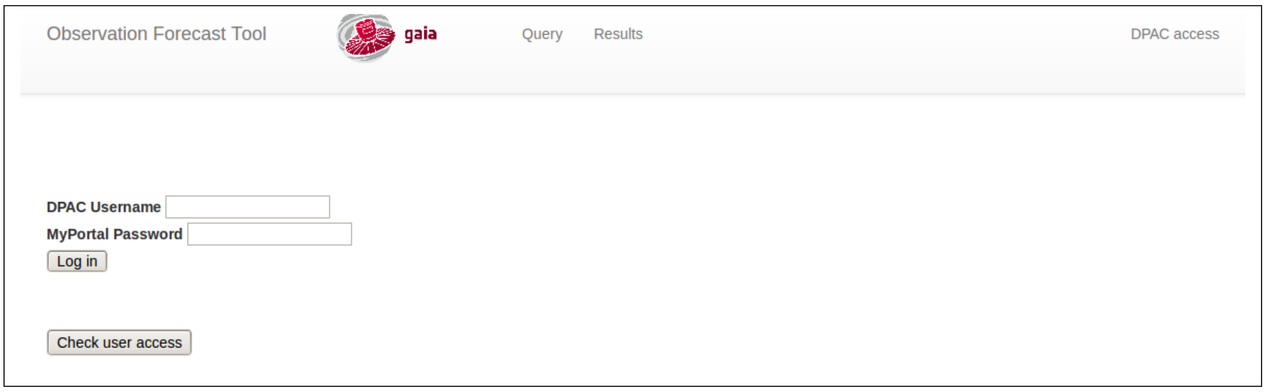
The result can be exported in different ways, each of them contain the time of the FPA transits potentially seen by Gaia. This is not a full guarantee of getting the transits observed. The target(s) could have been crossing the FoVs but for different reasons, the observed transits didn't get downlinked and/or processed.

In routine phase, GOST will compute the transit crossing the FoVs based on the routine nominal scanning law.

A different page - `properties.jsp` - will show information regarding the scan law used to compute the resulting time events. Other extra added information should be found there.

7.1 Login tab

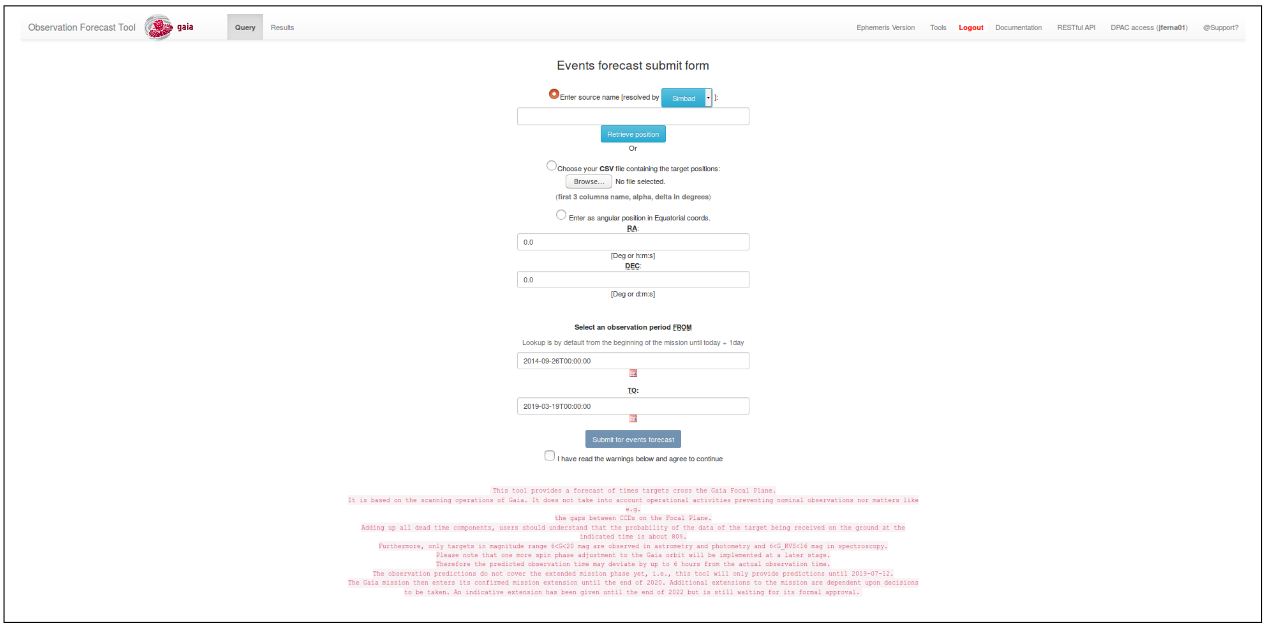
DPAC access page (`auth.jsp`) displays a login username and password fields to authenticate the DPAC user and enable full details output mode. Figure 4 displays the main authentication webform. The authentication required MyPortal credentials. The authenticated user will be redirected automatically to the home page (`index.jsp`) with the extra tabs (see Figure 5 and the details provided in the following section) as well as its username in bold format indicating full GOST details output mode is enabled.



DPAC Username

MyPortal Password

FIGURE 4: Authentication page.



Events forecast submit form

[✓ Enter source name [resolved by Simbad]:]

Or

Choose your **CSV** file containing the target positions:

No file selected.

(first 3 columns name, alpha, delta in degrees)

Enter as angular position in Equatorial coords.

RA: 0.0

[Deg or h:m:s]

DEC: 0.0

[Deg or d:m:s]

Select an observation period FROM

Lookup is by default from the beginning of the mission until today + 1day

2014-09-26T00:00:00

TO:

2019-03-19T00:00:00

I have read the warnings below and agree to continue

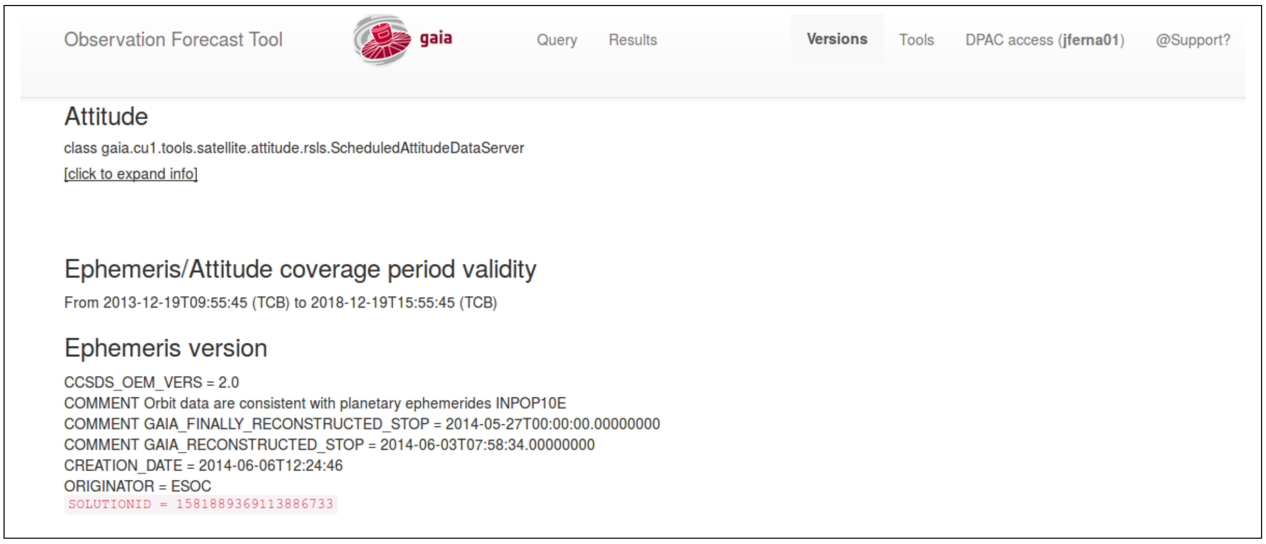
```
This tool provides a forecast of times targets cross the Gaia Focal Plane.
It is based on the scanning operations of Gaia. It does not take into account operational activities preventing nominal observations nor matters like
e.g.
the gaps between CCDs on the Focal Plane.
Adding up all dead time components, users should understand that the probability of the data of the target being received on the ground at the
indicated time is about 80%.
Furthermore, only targets in magnitude range 6<G<20 mag are observed in astrometry and photometry and 6<G_RVS<16 mag in spectroscopy.
Please note that one more spin phase adjustment to the Gaia orbit will be implemented at a later stage.
Therefore the predicted observation time may deviate by up to 6 hours from the actual observation time.
The observation predictions do not cover the extended mission phase yet, i.e., this tool will only provide predictions until 2019-07-12.
The Gaia mission then enters its confirmed mission extension until the end of 2020. Additional extensions to the mission are dependent upon decisions
to be taken. An indicative extension has been given until the end of 2022 but is still waiting for its formal approval.
```

FIGURE 5: Main page for an authenticated user.

7.1.1 versions tab

As we have pointed out before, once the user successfully login the system, there will a new version tab available. Figure 6 shows the versions page. It displays basic information for:

- the attitude data server. It shows the main class in GaiaTools that should be used in order to get access to the attitude. Clicking on "[click to expand info]" more details about the attitude is displayed (see Figure 7);
- the Ephemeris/Attitude coverage period valid. It provides the time range for the for which the Ephemeris/Attitude is valid;
- specific information for the ephemeris.



Attitude

class gaia.cu1.tools.satellite.attitude.rsls.ScheduledAttitudeDataServer

[click to expand info]

Ephemeris/Attitude coverage period validity

From 2013-12-19T09:55:45 (TCB) to 2018-12-19T15:55:45 (TCB)

Ephemeris version

```
CCSDS_OEM_VERS = 2.0
COMMENT Orbit data are consistent with planetary ephemerides INPOP10E
COMMENT GAIA_FINALLY_RECONSTRUCTED_STOP = 2014-05-27T00:00:00.00000000
COMMENT GAIA_RECONSTRUCTED_STOP = 2014-06-03T07:58:34.00000000
CREATION_DATE = 2014-06-06T12:24:46
ORIGINATOR = ESOC
SOLUTIONID = 1581889369113886733
```

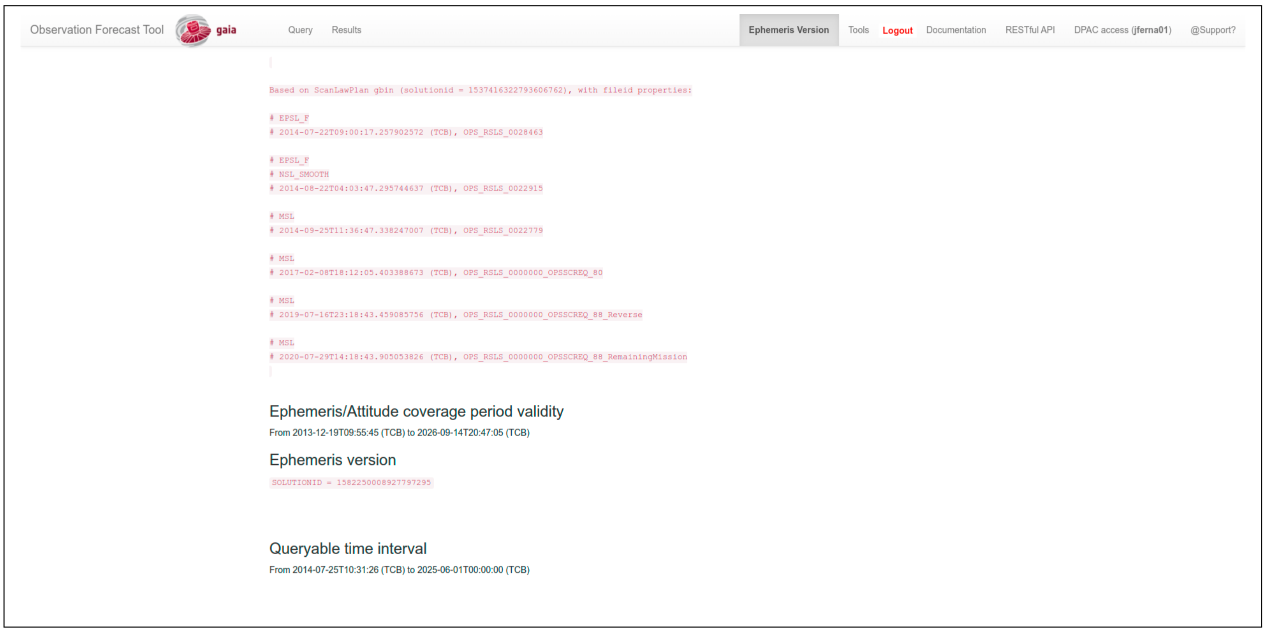
FIGURE 6: Versions page.

7.1.2 Tools tab

The Tools tab provides extra handy functionalities for the satellite. Figure 8 displays the main Tools page. As can bee seen, it contains the following options.

- *Where is Gaia now?* Clicking in this button downloads the file *gaia_live.kml*[8]. It can be opened with Google Earth and it will get position of Gaia (plus the distant to Earth) each 5 seconds.
- The *What is Gaia AOCS mode now?* button displays the Attitude and Orbit Control Sub-system (AOCS) mode used by Gaia. An example of the result is displayed in Figure 9. The possible available modes are *Stand-By Mode*, *Sun Acquisition Mode*,

[8]Keyhole Markup Language (KML) is an XML notation for expressing geographic annotation and visualization within Internet-based, two-dimensional maps and three-dimensional Earth browsers. KML was developed for use with Google Earth.



```
Based on ScanLawPlan gbin (solutionid = 1537416322793606762), with fileid properties:

# EPSL_F
# 2014-07-22T09:00:17.257902572 (TCB), OPS_RSLS_0028463

# EPSL_F
# NSL_SMOOTH
# 2014-08-22T04:03:47.295744637 (TCB), OPS_RSLS_0022915

# MSL
# 2014-09-25T11:36:47.338247007 (TCB), OPS_RSLS_0022779

# MSL
# 2017-02-08T18:12:05.403388673 (TCB), OPS_RSLS_0000000_OPSSCREQ_80

# MSL
# 2019-07-16T23:18:43.459085756 (TCB), OPS_RSLS_0000000_OPSSCREQ_88_Reverse

# MSL
# 2020-07-29T14:18:43.905053826 (TCB), OPS_RSLS_0000000_OPSSCREQ_88_RemainingMission
```

Ephemeris/Attitude coverage period validity

From 2013-12-19T09:55:45 (TCB) to 2026-09-14T20:47:05 (TCB)

Ephemeris version

```
SOLUTIONID = 1582250008927797295
```

Queryable time interval

From 2014-07-25T10:31:26 (TCB) to 2025-06-01T00:00:00 (TCB)

FIGURE 7: Expand version of the Version page, where the used scan law plan is displayed.

Inertial Guidance Mode, *Orbit Control Mode*, *TranSition Mode*, *Normal Mode*, and *Not defined*.

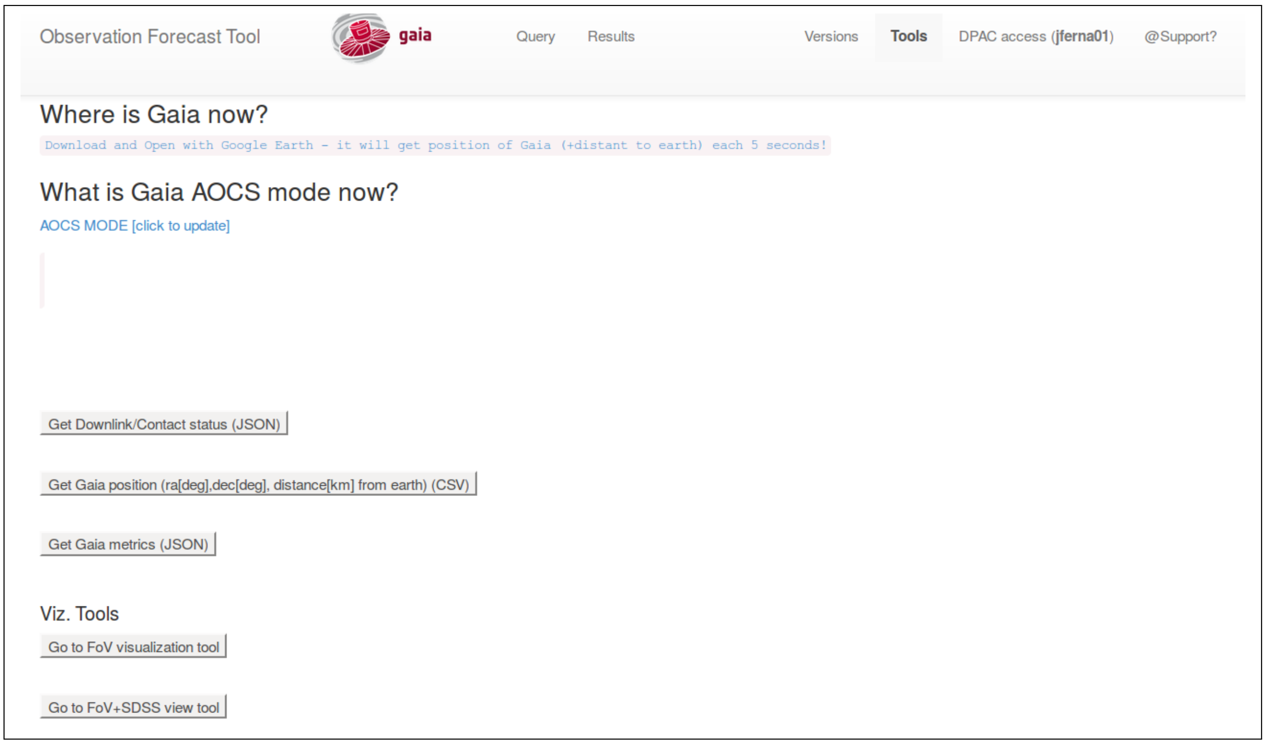
- The *Get Downlink/Contact status (JSON)* button generates the Downlink/Contact status in json format
- The *Get Gaia position (ra[deg], dec[deg], distance[km] from Earth) (CSV)* provides the right ascension and declination, in degrees, as well as the distance from Earth of the satellite in km. The information is displayed as comma-separated values (csv).
- The *Get Gaia metrics (JSON)* button displays the Gaia metrics in json format. This tool redirect the user to the service `https://gaia.esac.esa.int/missionstats/stats`.
- The *Go to FoV visualization tool* button displays the fields of views (FOV) of the satellite making use of the VirtualSky application[9]. The following are the main options that can be selected by the user:
  - The Gaia FOV;
  - the rate for the data updating;

[9]VirtualSky a browser-based planetarium from LCOGT that lets you see what is visible in the sky from any location on Earth. `http://http://virtualsky.lcogt.net/`

  - the initial UTC time. The button *Set time to now* sets the time to the present UTC time;
  - and the time step.

  In order to start the application, the user must press the *play* button. A blue or yellow square symbol for the Preceeding FOV or Following FOV, respectively, will appear at the sky image with its corresponding date, representing the position of the satellite. The user can zoom in/out the displayed image making use of the mouse. Note that the user can get extra information pressing keyboard '1'. The *pause* button stops the application.

- The *Go to FoV+SDSS view tool* button displays similar information as the previous one, but it also shows the sky observed by the SDSS catalogue (upper picture in Figure 11), making use of Aladin Lite[10]. Note that it is possible to select a different astronomical catalogue making use of the drop down menu of the Aladin lite window. The options that can be selected by the user are similar to the ones previously described.


Observation Forecast Tool
gaia
Query
Results
Versions
Tools
DPAC access (jferna01)
@Support?

Where is Gaia now?

Download and Open with Google Earth - it will get position of Gaia (+distant to earth) each 5 seconds!

What is Gaia AOCS mode now?

AOCS MODE [click to update]

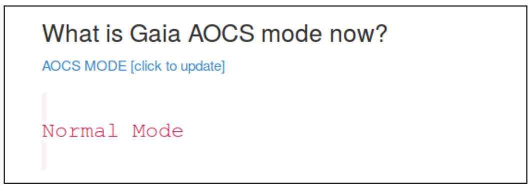
Get Downlink/Contact status (JSON)
Get Gaia position (ra[deg],dec[deg], distance[km] from earth) (CSV)
Get Gaia metrics (JSON)

Viz. Tools

Go to FoV visualization tool
Go to FoV+SDSS view tool

FIGURE 8: Tools page.

[10]Aladin lite is a lightweight version of the Aladin Sky Atlas, running in the browser and geared towards simple visualization of a sky region. `http://aladin.u-strasbg.fr/AladinLite/doc/`.

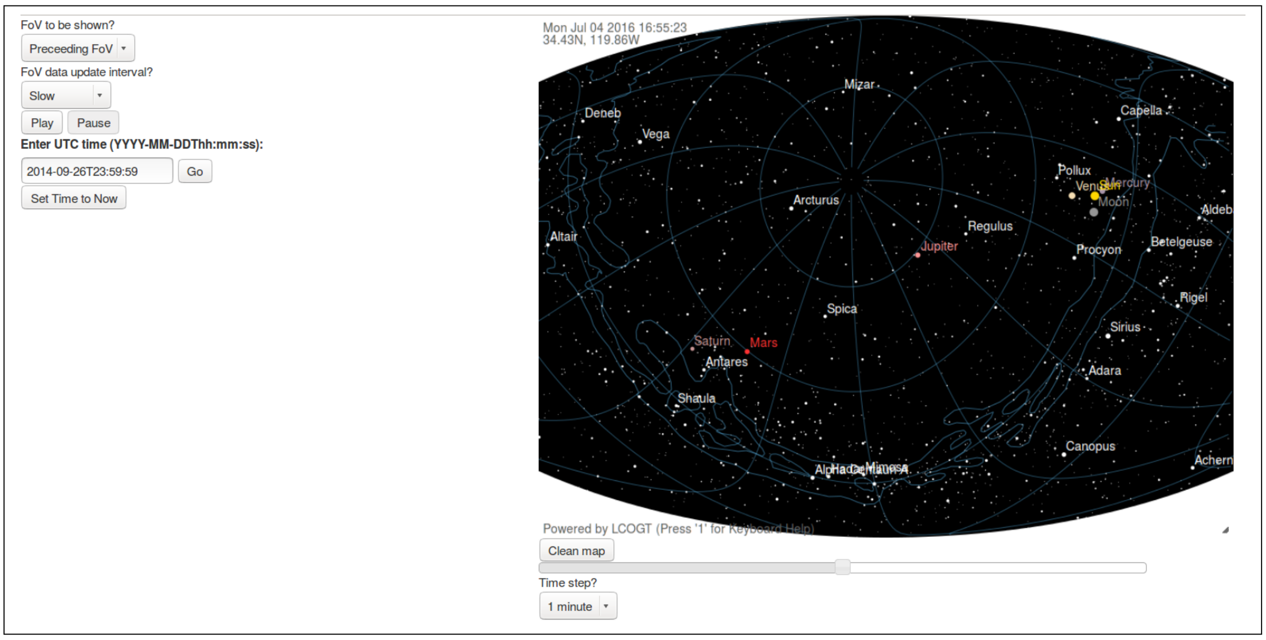


What is Gaia AOCS mode now?

AOCS MODE [click to update]

```
Normal Mode
```

FIGURE 9: Result of the *What is Gaia AOCS mode now?* button.



FoV to be shown? Preceeding FoV

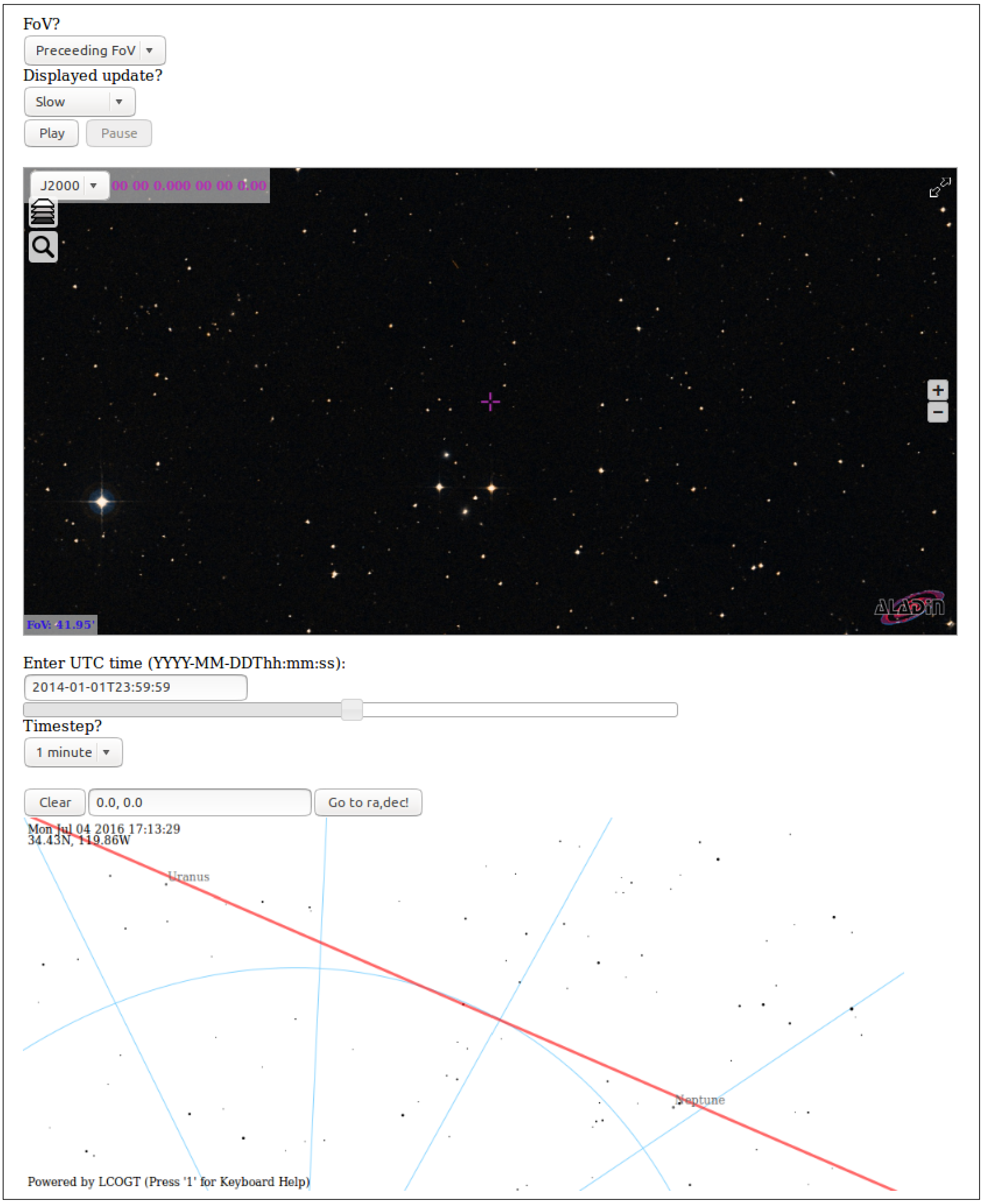
FoV data update interval? Slow

Enter UTC time (YYYY-MM-DDThh:mm:ss): 2014-09-26T23:59:59

Mon Jul 04 2016 16:55:23
34.43N, 119.86W

Time step? 1 minute

FIGURE 10: FoV visualization tool page.



FoV?

Preceeding FoV

Displayed update?

Slow

Play Pause

Enter UTC time (YYYY-MM-DDThh:mm:ss):

2014-01-01T23:59:59

Timestep?

1 minute

Clear 0.0, 0.0 Go to ra,dec!

FIGURE 11: FoV visualization tool and SDSS page.

.

8 Command line interface

GOST predictions can be obtained from CLI. The syntax of the script is as follows:

```
    Usage: bin/runGost.sh
                 [-h|--help]
                 [(-a|--attitude) <attitude>]
                 [(-s|--service) <service>]
                 [(-n|--name) <name>]
                 [(-r|--ra) <ra>] [(-d|--dec) <dec>] [-o|--torad]
                 [(-f|--tfrom) <tfrom>] [(-t|--runto) <runto>]
                 [(-i|--filein) <filein>]

  [-h|--help]
        Print this message

  [(-a|--attitude) <attitude>]
        Attitude data server can be: NSL37, EPSL, EpslAndNsl, MSL, SCHED or
           COMB

  [(-f|--tfrom) <tfrom>]
        Covered attitude from this time [ex.: J2012.0 or
        2012-01-01T12:00:00.000000000 (TCB) (default: J2014.0)

  [(-t|--runto) <runto>]
        RSLS coverage time duration [i.e. 5yr] (default: 5.0yr)

  [(-i|--filein) <filein>]
        Input file containing at least first 3 columns such as name, alpha,
        delta in radians (if in deg, use flag to convert)

  [-o|--torad]
        If present, it will convert the input angle in radians

  [(-r|--ra) <ra>]
        ICRS (EQU) RA in rad (default: 0.0)

  [(-d|--dec) <dec>]
        ICRS (EQU) DEC in rad (default: 0.0)

  [(-s|--service) <service>]
        Name resolver service - could be starting capital letter (V)izier,
        (N)ED, (S)imbad or (A)ll (default: )

  [(-n|--name) <name>]
        Name to be resolved (default: m31)
```

In order to execute the previous command the GOST package must be downloaded from the Gaia subversion service and the jar, including all the dependencies, created using the 'jarComplete' ant target.

The local gost properties are to be found where you define the -Dgaia.cu1.tools.util.prop.propdirs property flag. In the above script, the env.gost.properties has to be found under local folder name 'envprops'.

Example of env.gost.properties:

```
# Ephemeris data store provider
gaia.cu1.tools.data.GenericDataProvider=gaia.cu1.tools.data.GBinStoreDataProvider
gaia.cu1.tools.data.GBinStoreDataProvider.storeLocation=./data/eph

# IDTFLDB MIRROW
#gaia.cu1.tools.data.GenericDataProvider=gaia.cu1.tools.data.JdbcStoreDataProvider
#gaia.cu1.tools.data.JdbcStoreDataProvider.db.driver=com.intersys.jdbc.CacheDriver
#gaia.cu1.tools.data.JdbcStoreDataProvider.db.url=jdbc:Cache://gaials170.n1data.lan:56772/IDTFLOPSMIRROR
#gaia.cu1.tools.data.JdbcStoreDataProvider.db.username=********
#gaia.cu1.tools.data.JdbcStoreDataProvider.db.password=********

# Define attitude server to be used.
gaia.cu1.tools.satellite.attitude.AttitudeDataServer=gaia.cu1.tools.satellite.attitude.rsls.ScheduledAttitudeDataServer
#gaia.cu1.tools.satellite.attitude.AttitudeDataServer=gaia.cu1.tools.satellite.attitude.impl.numerical.
      CombinedAttitudeDataServer
#gaia.cu1.tools.satellite.attitude.rsls.ScheduledAttitudeDataServer
#gaia.cu1.tools.satellite.attitude.impl.analytical.EpslAndNsl

gaia.cu1.tools.satellite.attitude.AttitudeDataServer.location=./conf/
      OPS_RSLS_0022916_rsls_nsl_gareq1_afterFirstSpinPhaseOptimization.2.xml

# If EPSLAndNSL att server used, please configure these props too:
gaia.cu1.tools.satellite.attitude.AttitudeDataServer.switch=30d
gaia.cu1.tools.satellite.attitude.AttitudeDataServer.tStart=2012-01-01T12:00:00.000000000 (TCB)
gaia.cu1.tools.satellite.attitude.AttitudeDataServer.duration=5yr

# If CombinedServer att server is used, it needs:
gaia.cu1.tools.satellite.attitude.impl.numerical.IncrementalCombinedAttitudeDataServer.attitude=gaia.cu3.idtfl.idt.dm.Ioga gaia
      .cu1.mdb.cu3.agis.dm.Oga3 gaia.cu1.mdb.cu3.fl.dm.Oga2 gaia.cu1.mdb.cu3.idt.interm.dm.Oga1
gaia.cu1.tools.satellite.attitude.impl.numerical.CombinedAttitudeDataServer.attitude=gaia.cu3.idtfl.idt.dm.Ioga
#gaia.cu1.mdb.cu3.agis.dm.Oga3 gaia.cu1.mdb.cu3.fl.dm.Oga2 gaia.cu1.mdb.cu3.idt.interm.dm.Oga1

gaia.cu1.ost.service.GostServiceClient=gaia.cu1.ost.service.sesame.SesameClient

#gaia.cu1.ost.utils.scantype=DEFAULT
gaia.cu1.ost.utils.scantype=FULL
#FASTER

agis.source.SourceDirection=SourceDirectionGremImpl

gaia.cu1.tools.gost.web.startTime=2014-09-26T00:00:00
#2013-12-20T00:00:00

# Java temp folder
# (used in the gost reports to save memory)
java.io.tmpdir=.
```

Before execute this command we should provide the following inputs:

- Define the input file 'INPUT_ASCII', ASCII format, three comma-separated columns: Name,RA(deg),Dec(deg) like

  ```
  HIP104382,317.200797446,-88.9564792381
  HIP104385,317.197826281,-76.2125385922
  HIP104395,317.238948184,-5.5820777019
  ```

- START_ISO_DATE: Start ISO date of prediction UTC - i.e. '2014-01-02T12:00:00.000'
- DURATION : Duration in days/months.years/etc of the predicted period - i.e. 30d would be 30 days after Start date above

Taking these parameters into account, the resulting command would be:

```
$ bin/runGost.sh --tfrom <START_ISO_DATE> --runto <DURATION>
                 --torad --filein <INPUT_ASCII>
```

Define the configuration scanning law for predicted events to be used by setting the following properties

```
#Define attitude
gaia.cu1.tools.satellite.attitude.AttitudeDataServer=gaia.cu1.tools.satellite.attitude.rsls.SequenceAttitudeDataServerFactory
gaia.cu1.ost.generator.utils.attitude.scanlawplan=/mdb_scratch/data/latest/ScanLawPlan_000-000-000_file0.gbin
```

The output will be result.csv and .xml files. See Appendix A for more details about XML format.

9 Stand-alone version

The Gost package contains a stand-alone execution mode, that does not require any web server for its access. In this case the communication to the service is provided through a REST API[11].

Next we describe the main steps that have to be followed for its deployment and execution in a controlled environment:

- Configuration of the properties files.

  The property file located in the *conf-env* and passed to the script (for example, *conf-env/gost.webserver.properties* in the next point), must contain the following compulsory property:

  ```
  gaia.cu1.ost.progs.StandaloneServerPort=8080
  ```

  that defines the port that has to be used by any Rest client that tries to get access to the application.

  It is also necessary to define the following properties with the recommended values shown in order to launch the application in the operational environment[12]:

  ```
  # ------------------------------
  # Ephemeris data store provider
  # ------------------------------

  gaia.cu1.tools.data.GenericDataProvider=gaia.cu1.tools.data.GBinStoreDataProvider
  gaia.cu1.tools.data.GBinStoreDataProvider.storeLocation=/mdb_scratch/data/latest/

  # -----------------------------------
  # Define attitude server to be used.
  # -----------------------------------
  gaia.cu1.tools.satellite.attitude.AttitudeDataServer=gaia.cu1.tools.satellite.attitude.rsls.
      SequenceAttitudeDataServerFactory
  gaia.cu1.ost.generator.utils.attitude.scanlawplan=/mdb_scratch/data/latest/ScanLawPlan_000-000-000_file0.gbin


  # If EPSLAndNSL att server used, please configure these props too:
  gaia.cu1.tools.satellite.attitude.AttitudeDataServer.switch=30d
  gaia.cu1.tools.satellite.attitude.AttitudeDataServer.tStart=2012-01-01T12:00:00.000000000 (TCB)
  gaia.cu1.tools.satellite.attitude.AttitudeDataServer.duration=10yr

  #Plan view file - should be updated version from svn!
  gaia.cu1.ost.service.metrics.extras.plnview=/lhome/gostops/GOST/conf/planview.txt

  # If CombinedServer att server is used, it needs:
  gaia.cu1.tools.satellite.attitude.impl.numerical.IncrementalCombinedAttitudeDataServer.attitude=gaia.cu3.idtfl.
      idt.dm.Ioga gaia.cu1.mdb.cu3.agis.dm.Oga3 gaia.cu1.mdb.cu3.fl.dm.Oga2 gaia.cu1.mdb.cu3.idt.interm.dm.Oga1
  gaia.cu1.tools.satellite.attitude.impl.numerical.CombinedAttitudeDataServer.attitude=gaia.cu3.idtfl.idt.dm.Ioga

  # -----------------------
  # Forecast configuration
  # (Except scan law setup)
  ```

[11] At the time of writing this document, the access to this stand-alone version is restricted to connections done within ESAC

[12] Note that the properties that define paths to files have to be reviewed by the operators, since it could be required to update these files as soon as new versions were available. In particular the planview.txt represents a link to the file https://gaia.esac.esa.int/dpacsvn/DPAC/GaiaOperations/MOCSOC/PlanView/PLNVIEW_20200604T000000_20210101T000000_20200604T120025_v0100.PASSO.consolidated.txt, that is regularly updated in the repository. This file is used by the rest resource /gaiaplanview. So in order to make use of an up-to-date and synchronized file, an additional mechanism should be in place.

```
# -----------------------

gaia.cu1.ost.lookup.GostLookupServer=gaia.cu1.ost.web.InMemoryScanner

gaia.cu1.ost.utils.scantype=FULL

agis.source.SourceDirection=SourceDirectionGremImpl

# Increate the valid search range
gaia.cu1.tools.gost.web.startTime=2014-07-25T10:31:26
gaia.cu1.tools.gost.web.workers=10

# Disable these services so regular tasks can finish
gaia.cu1.tools.data.timeline.TimeLineAccumulator.enabled=false
gaia.cu1.tools.util.solutionid.inputdata.InputDataTracker.enabled=false

gaia.cu1.tools.gost.web.maxnumbersources=10000
```

- Launch the application.

  The application is launched making use of the script */bin/launchGostStandAlone.sh*:

  ```
  ./bin/launchGostStandAlone.sh -p conf-env/gost.webserver.properties
  ```

  where the flag *-p* defines the property file to be loaded (usually located in conf-env/). The optional flag *-j* can be used to pass parameters to the jvm, such as the maximum heap memory used by the application. For example:

  ```
  ./bin/launchGostStandAlone.sh -p conf-env/gost.webserver.properties
      -j "-Xmx10g "
  ```

  If this flag is not set, a default heap memory of 2 gigabytes would be used. The flag *-h* prints the help message.

  The script reads also the property file */conf/logging.properties*, that defines the characteristics of the output log file. Note also that the script uses a maximum memory of 2 GB.

  The script will generate a log file with the following name */logs/GostStandAlone-LOGDATE.log*, where LOGDATE represents the execution date of the script, for example */logs/GostStandAlone-2019-11-15T15:11:16.log*.

- Checking the application.

  Once the system is up and running, and in case no error is found in the log file, any client would have access to the service through the url

  gaialsXX.n1data.lan:8080/gost/rest,

  where *gaialsXX.n1data.lan* represents the name of the machine where the application was launched, and the value *8080* is the port previously opened that was defined in the property file.

  Also, it is possible to check the services provided by the stand-alone application by the execution of the script

  ./bin/testRestServices.sh -u url -o outputdir,

  where *-u* defines the machine where the gost is running, for example *https://localhost:8080* or *https://gaia.esac.esa.int/vgost* and *-o* the output directory where intermediate results are saved. If this flag is not set, the *log* file would be used.

- Stop of the application.

  The application is stopped by making use of the linux command *kill*.

10 Comparing Gost forecasts

When the configuration or execution environment of the GOST is changed, it is needed a way to evaluate if the results from GOST have changed and if the changes are 'reasonable'.

'Reasonable' may mean that they are within the expected accuracy of GOST and can be understood based on the differences in the execution environment (e.g. expected changes in trigonometric precision between Java versions).

The bash scrip `CompareGostResults.sh` accepts as input the output from two different executions of GOST and generates a set of statistics on the differences between the two outputs. These statistics can be used to check if the differences between the two GOST outputs are 'reasonable', treating the GOST as a black box.

The program accepts as input the following two possible formats:

1. the GOST output in the csv format for a given input list of sources, start time and end time. In this case it accepts the extended DPAC and the 'public' formats;
2. the GOST output in the xml format for a given input list of sources, start time and end time. This information is only available for DPAC members.

The program generates three outputs:

1. a log file in the log directory with the format *CompareGostResults-yyyy-mm-ddTHH:MM:SS.log*;
2. per each target in the input file, a csv file with the comparison of the variables available in the input files. The format of the output file is
   compareGostResults_TYPE_TARGETNAME_yyyy-mm-ddTHH:MM:SS.csv,
   where *TYPE* identifies the format of the input files, xml, csv_dpac or csv_public, and *TARGETNAME* is the name of each target contained in the first input file.
3. a txt file with the format *CompareGostResults_report_TYPE_yyyy-mm-ddTHH:MM:SS.txt* that includes
   - the number of unique sources observed per file;
   - the number of observations in total observed;
   - the matches between the observations in both files;
   - the numerical differences of the processed variables.

The script is executed by the command `./bin/CompareGostResults.sh`. The flag *-h* prints all the required parameters

```
./bin/CompareGostResults.sh  -h

Parameters:

-h|--help print help message
-f|--firstFile First input file (xml or csv)
-s|--secondFile Second input file (xml or csv)
-o|--outputDir Output dir
-c|--preciCoordRad Precision in the comparison of the coordinates [rad](
   optional)
-t|--preciTimeSec Precision in the comparison of the time [sec](optional)
```

Only the three first flags are required, while the last two are optional. The flag *-c* sets the upper limit in the comparison of the coordinates of the target of the input files, while the flag *-t* sets the upper limit in the comparison of the TCB times for the xml and cvs_dpac types, and the TcbBarycentricJulianDateAtBarycentre times for csv_public types. The default values are $1.0E - 9$.

11 Command line validation

The following three scripts available in bin folder are used for validating a release candidate with the following purposes:

1. The testQueryVisObjService.sh script is used for testing the correct access to Object Visibility Simple Access Protocol exposed by Gost.
2. The testRestServices.sh script is used for testing the rest services exposed by Gost.
3. The validateGostResults.sh script is used for ensuring determinism of GOST predictions. It compares the gost forecast file (csv file) for a reference input catalogue, w.r.t a reference Gost forecast file (available in cu1 ftp data, generated with Gost 22.1.1, tomcat 9.0.56, and MDB ephemeris data from 12/05/2022).

Each script is executed as follows:

1. bin/testQueryVisObjService.sh -u url
2. bin/testRestServices.sh -u url -o output folder
3. bin/validateGostResults.sh -u url -p property file

Examples of execution:

1. bin/testQueryVisObjService.sh -u https://gaia.esac.esa.int/val_gost/
2. bin/testRestServices.sh -u https://gaia.esac.esa.int/val_gost/
3. bin/validateGostResults.sh -u https://gaia.esac.esa.int/val_gost/ -p conf-env/validateresults.properti

The script validateGostResults.sh shall be executed twice:

1. The first execution to validate the past range of the mission time (2014 to 2022), with tolerances values 0. Time range (start and end) and tolerances are defined in conf-env/validateresults.properties
2. The second execution to validate the future mission time range (2022-2025), with higher tolerances. Time range (start and end) and tolerances are defined in conf-env/validateresults.properties

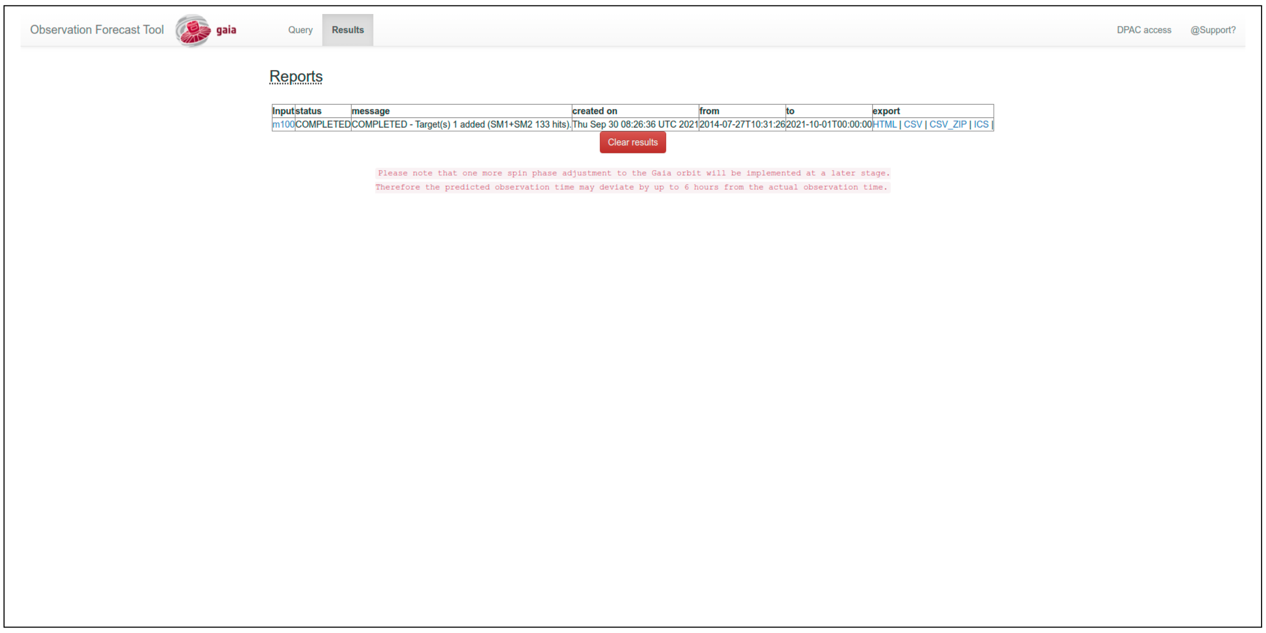
12 References

[GAIA-LL-061-4], Lindegren, L., Bastian, U., 2018, *Local plane coordinates for the detailed analysis of complex Gaia sources*,
`GAIA-LL-061-4`,
URL `https://dms.cosmos.esa.int/cs/livelink/Open/504573`

A Output format

The goal of this section is to explain the different format and in particular the XML structure as well as the different elements from a query result either from CLI or from web interface.

Figures 12 and 13 display the result page that a public user and a DPAC-user is redirected, respectively, after a query has been completed. Both of them show an export column which are links to the other output format.



Input	status	message	created on	from	to	export
m100	COMPLETED	COMPLETED - Target(s) 1 added (SM1+SM2 133 hits).	Thu Sep 30 08:26:36 UTC 2021	2014-07-27T10:31:26	2021-10-01T00:00:00	HTML | CSV | CSV_ZIP | ICS |

```
Please note that one more spin phase adjustment to the Gaia orbit will be implemented at a later stage.
Therefore the predicted observation time may deviate by up to 6 hours from the actual observation time.
```

FIGURE 12: Web page for one year prediction of m31 object, for a public user.

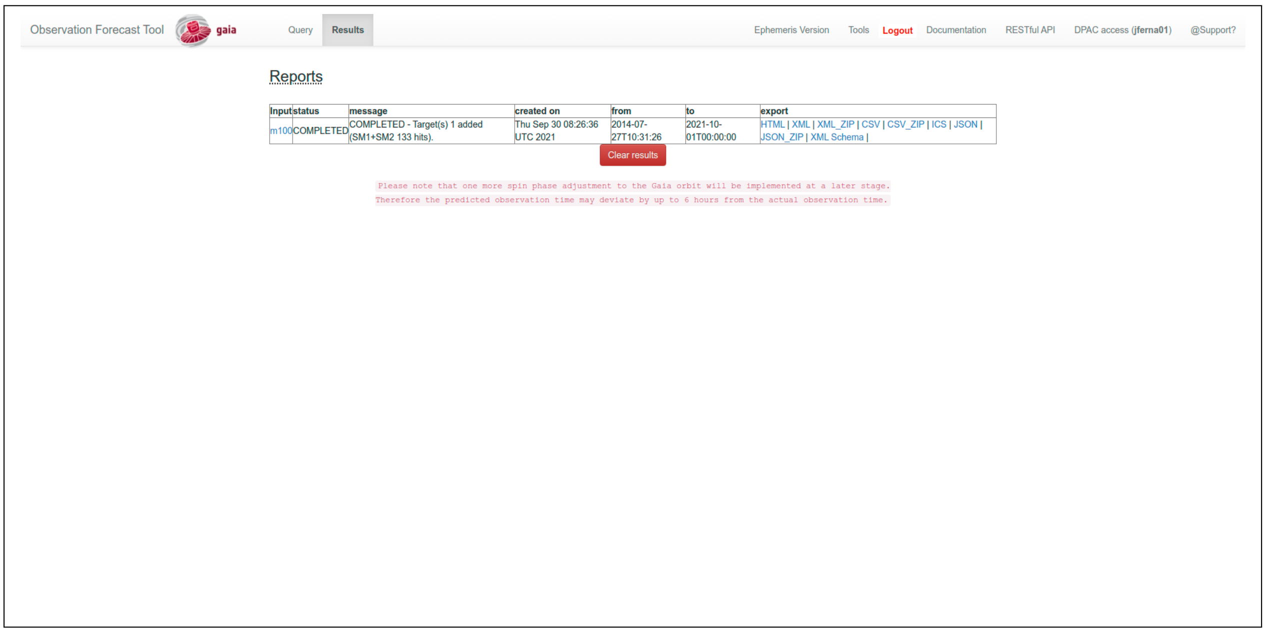
As can be seen on the right column, the results can be rendered in different formats: HTML, XML, CSV and ICS[13]. But, as can bee seen, only the DPAC-user has access to the xml file, and its associated xsd schema file, that contains all the available information of the requested object. Moreover, the output HTML and CSV files for the public user and the DPAC user contains different information. Table 3 summarizes the information provided by the csv file for each type of user.

For example the following is a CSV result for the target m31 in the interval 2014-09-26T00:00:00 to 2015-06-26T00:00:00 (see figure 1):

[13] An ICS file is a calendar file saved in a universal calendar format used by several email and calendar programs, including Microsoft Outlook, Google Calendar, and Apple Calendar. It enables users to publish and share calendar information on the web and over email.

TABLE 3: Description of the columns contained in the output csv for a public and a DPAC user.

column name	public	DPAC-only	units
Target	available	available	–
ra[rad]	available	available	[radians]
dec[rad]	available	available	[radians]
ra[h:m:s]	available	available	[h:m:s]
dec[d:m:s]	available	available	[d:m:s]
ObservationTimeAtGaia[TCB]	not available	available (all CCDs)	[TCB]
ObservationTimeAtGaia[UTC]	available for AF4	available (all CCDs)	[UTC]
CcdKind	not available	available (all CCDs)	[SM/AF]
CcdRow[	available for AF4	available (all CCDs)	[1-7]
CcdStrip	not available	available (all CCDs)	[SM1-AF9]
normalizedAcPosition	not available	available (all CCDs)	[0-1]
scanAngle	available for AF6	available (all CCDs)	[radians]
Fov	available for AF6	available (all CCDs)	[FovP=preceding/FovF=following]
CcdAcCoordinate]	not available	available (all CCDs)	[0-1966]
parallaxFactorAlongScan	available for AF4	available (all CCDs)	–
parallaxFactorAcrossScan	available for AF4	available (all CCDs)	–
ObservationTimeAtBarycentre	available for AF4	available (all CCDs)	[BarycentricJulianDateInTCB]
FovPcentreDirRa	not available	available (all CCDs)	[radians]
FovPcentreDirDec	not available	available (all CCDs)	[radians]
FovFcentreDirRa	not available	available (all CCDs)	[radians]
FovFcentreDirDec	not available	available (all CCDs)	[radians]



Reports

Input	status	message	created on	from	to	export
m100	COMPLETED	COMPLETED - Target(s) 1 added (SM1+SM2 133 hits).	Thu Sep 30 08:26:36 UTC 2021	2014-07-27T10:31:26	2021-10-01T00:00:00	HTML | XML | XML_ZIP | CSV | CSV_ZIP | ICS | JSON | JSON_ZIP | XML Schema |

```
Please note that one more spin phase adjustment to the Gaia orbit will be implemented at a later stage.
Therefore the predicted observation time may deviate by up to 6 hours from the actual observation time.
```

FIGURE 13: Similar to Fig. 12 but for a DPAC user.

```
Target, ra[rad], dec[rad], ra[h:m:s], dec[d:m:s], ObservationTimeAtGaia[UTC
    ], CcdRow[1-7], normalizedAcPosition[0-1], scanAngle[rad], Fov[FovP=
    preceding/FovF=following], parallaxFactorAlongScan,
    parallaxFactorAcrossScan, ObservationTimeAtBarycentre[
    BarycentricJulianDateInTCB]
m31
    ,0.1864833394501661,0.72027556568241,00:42:44.330,+41:16:07.500,2015-01-27
    T01:21:50.080 ,6,-0.8881621789904904,0.20540118433748625,FoVF
    ,0.6644223629978906,-0.013000669597906947,2457049.5571136656
m31
    ,0.1864833394501661,0.72027556568241,00:42:44.330,+41:16:07.500,2015-02-21
    T19:14:51.828 ,6,-0.8580562050512145,-1.4831668956182447,FoVF
    ,-0.09098013222035409,-0.6582828237594386,2457075.300256
m31
    ,0.1864833394501661,0.72027556568241,00:42:44.330,+41:16:07.500,2015-02-21
    T21:01:25.997 ,5,-0.2021439141779356,-1.4869791626290962,FoVP
    ,-0.09348786721012495,-0.6579312051514988,2457075.374257385
```

Event times are in TCB. CCD strip type (SM or AF), row number (1-7), strip number (1-11) 1 and 2 being SMs and 3 to 11 AFs, across scan normalized position (zeta angle) and the scan angle as defined in GAIA-LL-061-4.

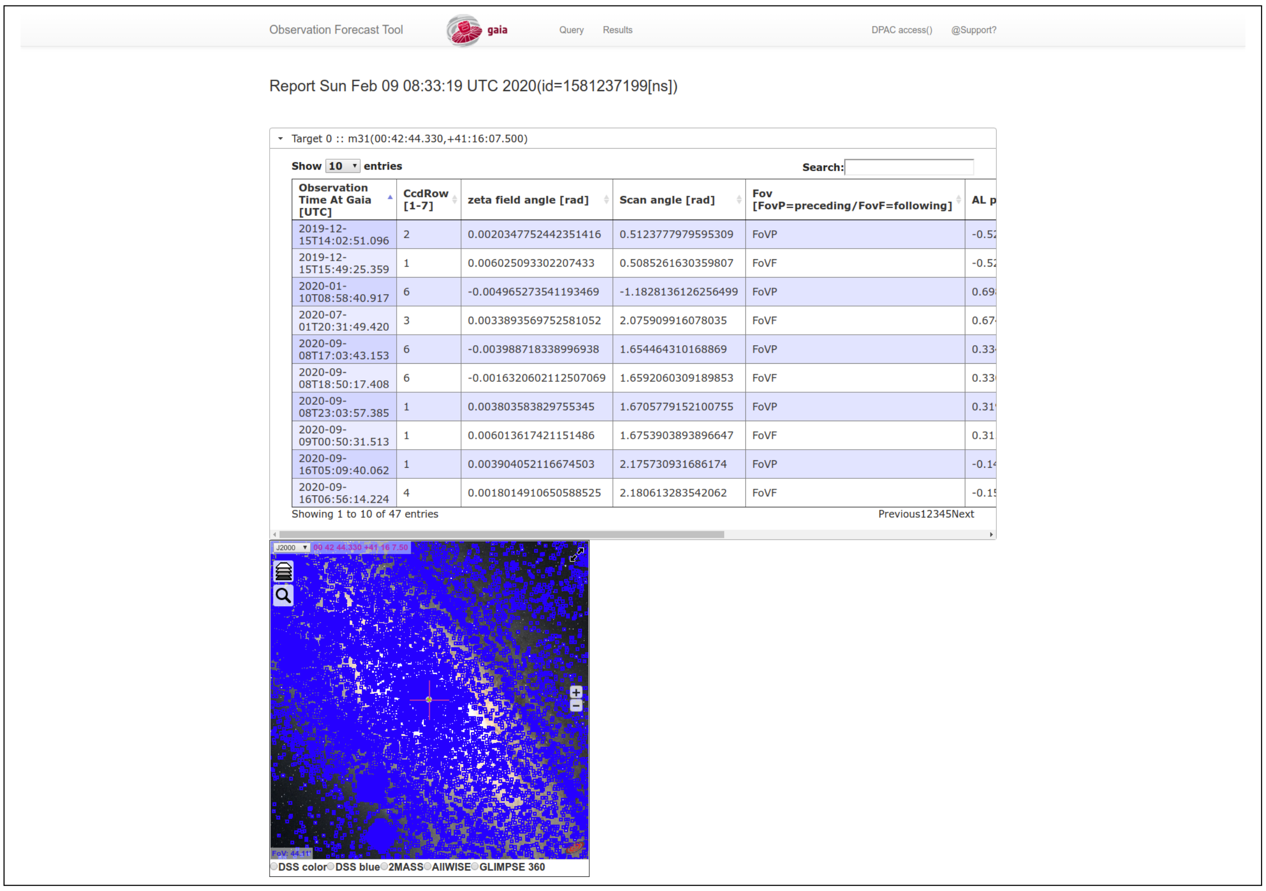
An example of an XML formatted output can be seen in section 4.3

- <creationTime >: the time when the report has been created

- <attitudeDataServerId >: information of the attitude data being used to make the prediction - usually the XML configuration file name used
- <startTime>and <endTime>which are the time range which covers the report of the predictions - in this case, 5 years from launch to 20th of June 2014.
- Then a list of <targets><target>, each of them with: <coords>ra dec and an identifier <id>which is generally a running number that make sense if more than one target has been asked in the prediction.
- Each <target>will contain a list of events element <event>
- Each of <event>contains:
  - <evenDate>: the predicted transit date
  - <details>: wrap details such as kind, row, strip numbers.
  - <angleToZaxis>: the angle of the target seen in Gaia SRS and Z axis
  - <acCcdCoordinate>: the across pixel position (pixel column)
  - <normAcTransitPosition>: the across normalized angle transit position
  - <scanAngle>: the scanning angle
  - <pfovdir>and <ffovdir>are the vector direction of the Preceeding and Following FoV in degrees and in ICRS frame

For the ICS format, the file in this output format can directly be used in any iCal reader client.

The HTML format displays the results in a table (see figure 14). As can be seen, the number of the displayed results can be selected by the left top button (the default number of values is 10). The information displayed in the table is basically the same information as the one is obtained in the XML format (see previous paragraph about the rendered XML). At the lower left corner, the image of the object is displayed making use of the 2MASS catalogue at epoch J2000. Both the catalogue and the epoch can be selected in the drop down menu of the figure. The figure is generated by Aladin Lite.



Report Sun Feb 09 08:33:19 UTC 2020(id=1581237199[ns])

Target 0 :: m31(00:42:44.330,+41:16:07.500)

Show 10 entries Search:

Observation Time At Gaia [UTC]	CcdRow [1-7]	zeta field angle [rad]	Scan angle [rad]	Fov [FovP=preceding/FovF=following]	AL p
2019-12-15T14:02:51.096	2	0.0020347752442351416	0.5123777979595309	FoVP	-0.52
2019-12-15T15:49:25.359	1	0.006025093302207433	0.5085261630359807	FoVF	-0.52
2020-01-10T08:58:40.917	6	-0.004965273541193469	-1.1828136126256499	FoVP	0.69
2020-07-01T20:31:49.420	3	0.0033893569752581052	2.075909916078035	FoVF	0.67
2020-09-08T17:03:43.153	6	-0.003988718338996938	1.654464310168869	FoVP	0.33
2020-09-08T18:50:17.408	6	-0.0016320602112507069	1.6592060309189853	FoVF	0.33
2020-09-08T23:03:57.385	1	0.003803583829755345	1.6705779152100755	FoVP	0.31
2020-09-09T00:50:31.513	1	0.006013617421151486	1.6753903893896647	FoVF	0.31
2020-09-16T05:09:40.062	1	0.003904052116674503	2.175730931686174	FoVP	-0.14
2020-09-16T06:56:14.224	4	0.0018014910650588525	2.180613283542062	FoVF	-0.15

Showing 1 to 10 of 47 entries Previous12345Next

FIGURE 14: html page for one year prediction of the m31 object in HTML format, for a DPAC user.