CLS
COLLECTE LOCALISATION SATELLITES

ARGOS

Argos Web Service Interface Specification

Reference:	CLS-DT-NT-10-165
Nomenclature:	ARG-IF-22-1427-CLS
Issue:	1. 4
Date:	**Mar. 19, 13**

CLS (siège)
8-10 rue Hermès
Parc technologique du Canal
31520 Ramonville Saint-Agne
FRANCE

Tél. : +33 (0)5 61 39 47 00
Fax : +33 (0)5 61 75 10 14
Mél. : info@cls.fr
Web : www.cls.fr

CLS Brest Le Ponant
Avenue La Pérouse
29280 Plouzané
FRANCE

Tél. : +33 (0)2 98 05 76 80
Fax : +33 (0)2 98 05 76 90

Chronology Issues:			
Issue:	Date:	Reason for change:	Author
0.0	15/04/10	Draft	M.Turiot
1.0	19/05/10	First version. Interface specification added. XML schema finalized	M.Turiot
1.1	03/12/10	KML is now supported XML <frequency> and <doppler> fixed	M.Turiot
1.2	20/05/11	Error ellipses available in KML Platform <platformHexId> available in XML	M.Turiot
1.3	07/02/12	Precision on protocols Requests by processing dates	M. Turiot
1.4	19/02/13	Access to users platform list Access to observation data	M. Turiot

People involved in this issue:		
Written by (*):	M. Turiot	Date + Initials:(visa or ref)
Checked by (*):	DT-AQM	Date + Initial:(visa ou ref)
Approved by (*):	C. Patou Y. Bernard	Date + Initial:(visa ou ref)
Application authorized by (*):		Date + Initial:(visa ou ref)

**In the opposite box: Last and First name of the person + company if different from CLS*

Index Sheet:	
Context:	Argos Processing Center
Keywords:	ARGOS DWS
Hyperlink:	www.argos-system.org/manual/webservices.pdf

Distribution:		
Company	Means of distribution	Names
CLS	Notification	Y.Bernard, A.Bes, B.Coulon, A.Fontanaud, E.George, M.Guigue, E.Lambert, C.Patou, B.Pirrotta, M.Sabatier, M.Turiot, S.Vincent

List of tables and figures

List of tables:

List of figures:

List of items to be confirmed or to be defined

Lists of TBC:

Lists of TBD:

Aucune entrée de table des matières n'a été trouvée.

Applicable documents

AD 1 Plan d'assurance produit de CLS
CLS-ED-NT-03-394

Reference documents

RD 1 Manuel du processus Documentation
CLS-DOC

List of Contents

1. Object

This document describes the Web Service interface provided by the Argos Processing Center (APC) system to users.

This document comprises the following chapters apart from this one:

- Chapter 2 gives the Provider and Consumer of the system and the exchange protocol for each interface
- Chapter 3 gives the description of each request and its response.
- Chapter 0 gives the description of the main objects used in requests and responses.
- Chapter 6 gives examples.

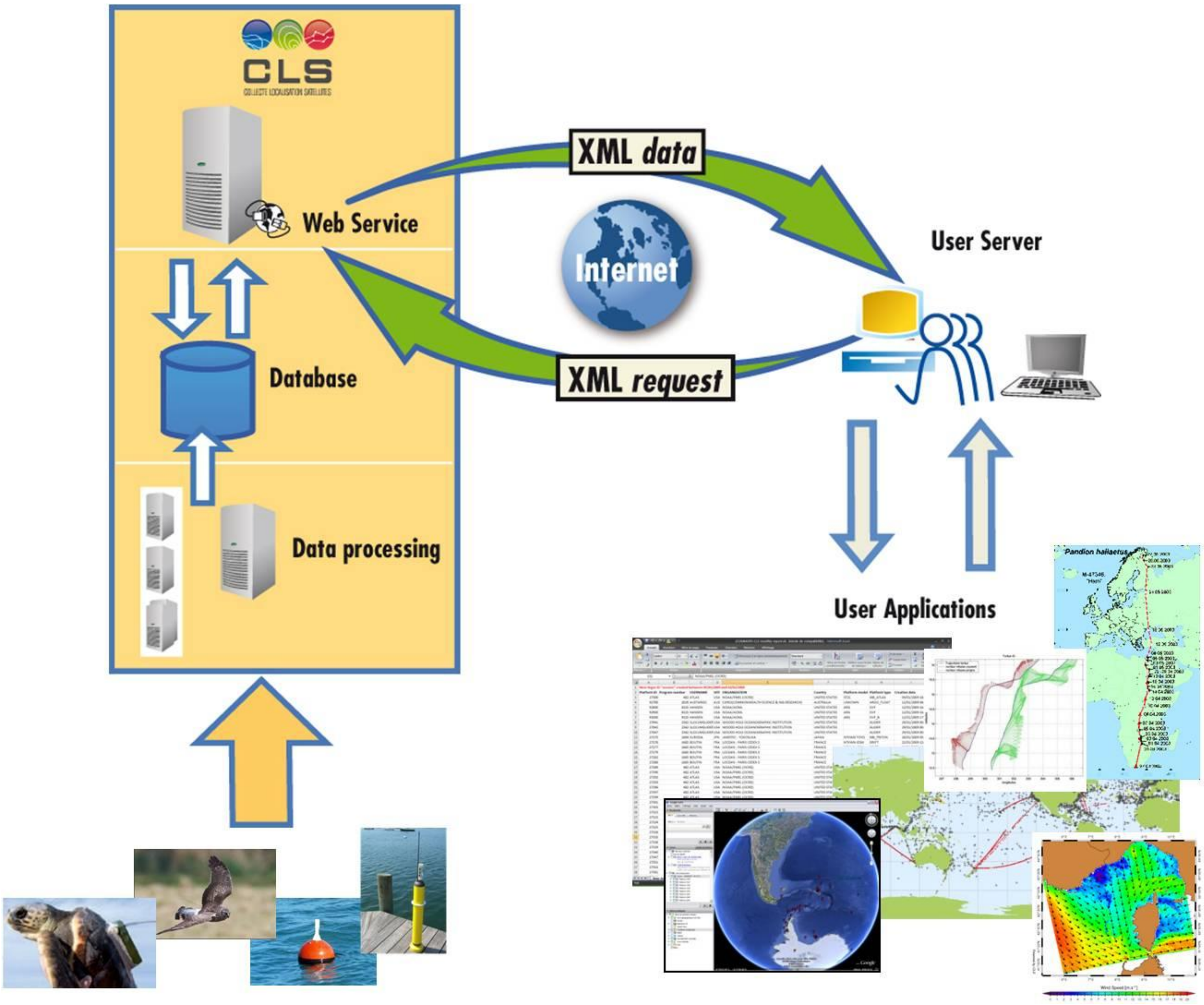


Figure 1= Argos Web Service Overview

2. General view of the interface

2.1. Definitions

Request: a request is a message coming from a user to the APC through the DWS Interface (Argos Webservice interface called DWS).

Response: in return to a request sent to the APC, it is a message answer to the user.

2.2. Interface identification

The DWS interface provides several services to users. These services are used to consult data recorded in the APC

The following table presents each command:

Identifier	Description
getXsd	*Interface provided by the APC to users to get the xml schema of the data returned by getXml and getStreamXml commands*
getXml	*Interface provided by the APC to users to get Argos data. This command should be used for small data sets (a few satellite passes)*
getStreamXml	*Interface provided by the APC to users to get Argos data. This command shall be used for larger data sets*
getCsv	*Interface provided by the APC to users to get Argos data in comma separated values*
getKml	*Interface provided by the APC to users to get Argos locations in kml format (Keyhole Markup Language)*
getObsXml	*Interface provided by the APC to users to get Observation data.*
getObsCsv	*Interface provided by the APC to users to get Observation data in comma separated values format.*
getPlatformList	*Interface provided by the APC to users to get the list of available platforms.*

Table 1: list of available commands

2.3. Interface protocol

The interface provided by the Argos Processing Center (APC) is a web service.

The protocol used to communicate between the APC and the user is based on **SOAP** (Simple Object Access Protocol) over **HTTP**.

The **SOAP** protocol allows exchanging data in XML format. This is an RPC (remote access protocol) object oriented in XML. The **SOAP** protocol can be used over HTTP, **HTTPS**, SMTP … (cf. http://www.w3.org/TR/soap/ for SOAP protocol description, http://www.w3.org/TR/soap12-mtom/ for MTOM)

This WebService uses SOAP Version 1.2, and in particular makes usage of MTOM (Message Transmission Optimization Mechanism) to transfer the data to the client.

The web service WSDL and actual address will be provided on request to CLS UserOffice.

2.4. Using the web service

2.4.1. Submitting a request

The calling system will call a service with the required parameters.

The result will be synchronously returned in the form of a string or an octet stream

2.4.2. Handling errors

Errors occurring as a result of invalid processing of the request are returned in the response data.

Unexpected errors which may occur as a result of any other error condition will be returned as SOAP faults with a message describing the error.

3. Web Service Interface Description

3.1. getXsd

This request is used to ask for the xml schema used by getStreamXml and getXml requests.

3.1.1. Parameters

There is no parameter.

3.1.2. Return value

The getXsdResponse is a string containing the xml schema of the getXml and getStreamXml responses. The present version of the xml schema is given in §4.2

3.2. getXml

This request is used to ask Argos satellite passes and detailed content.

3.2.1. Parameters

Authentication: required

- username
  authentication user name. Same as for present customer ArgosWeb and Telnet access.
- password
  authentication password. Same as for present customer ArgosWeb and Telnet access.

Required data selection: choice between

- platformId
  comma separated list of platforms.

- programNumber
  comma separated list of programs. All platforms of the program will be accessed.

Time selection: choice between

- startDate / endDate
  satellite pass best_date begin of search and end of search. Only the 10 last days are available. If endDate is not specified, the current date is assumed.
- nbDaysFromNow
  returns only satellite passes from the last days. Only the 10 last days are available.

Time reference: choice between

- BEST_MSG_DATE (default)
  satellite pass best_date is used for begin of search and end of search.
- MODIFICATION_DATE
  processing daty by the Argos Processing Center is used for begin of search and end of search.

Output size: optional

- nbPassByPtt
  maximum number of passes returned by platformId. If there are actually more satellite passes, only the first ones, with respect to best_date, will be returned to the user. In conjunction with mostRecentPassages option (see below) the last ones, with respect to best_date, will be returned. By default, all satellite passes will be returned.

Data filters: optional. Several filters may be used simultaneously.

- locClass
  List of satellite pass location classes returned. The Argos location classes are Z,B,A,0,1,2,3,G. By default, all satellite passes are returned.
- geographicArea
  [TBC]Zone identification. Only satellite passes in the zone are returned. By default, no zone restriction. Don't use, not implemented yet.
- compression
  Minimal compression. Only collect messages with compression value higher or equal to the required value will be returned. Impacts satellite passes and location. By default, all messages are returned.
- mostRecentPassages
  If this option is set, the most recent passages with respect to best_date are returned. Only the 10 last days are available. By default, the oldest passages with respect to best_date are returned.
- argDistrib
  By default, sensors with Argos distribution flag are returned. This option allows returning other sensors, and is mainly for CLS internal use.

Output content: optional. Several filters may be used simultaneously.

By default, without any flag set, only the location is returned: program, platform, satellite_pass, location, except <latitude2>, <longitude2>, <altitude2> elements

- displayLocation
  Data from the following sequences are returned: program, platform, satellite_pass, location, except <latitude2>, <longitude2>, <altitude2> elements
- displayDiagnostic
  Data from the following sequences are returned: program, platform, satellite_pass, location, diagnostic, except <latitude2>, <longitude2>, <altitude2> elements
- displayMessage
  Data from the following sequences are returned: program, platform, satellite_pass, message

- displayCollect
  Data from the following sequences are returned: program, platform, satellite_pass, message, collect, except <raw_data> element
- displayRawData
  Data from the following sequences are returned: program, platform, satellite_pass, message, collect, with <raw_data> element
- displaySensor
  Data from the following sequences are returned: program, platform, satellite_pass, message, format, sensor
- displayImageLocation
  Data from the following sequences are returned: program, platform, satellite_pass, location, with <latitude2>, <longitude2>, <altitude2> elements. Those location flags only apply to platforms using Least Squares location algorithm.
- displayHexId
  with this flag set (true), the platform hexadecimal identifier is returned in the platform data

3.2.2. Return value

The getXmlResponse is a string containing the data. The data are compliant with the xml schema returned by getXsd.

The details of the xsd elements is given in chapter 4.2

The result is truncated to 100 satellite passes. This limit may be changed by CLS.

If there are actually more data than the returned data, the following error is returned with the data:

```
<errors>
 <error code="2">max response reached</error>
</errors>
```

3.3. getStreamXml

This request is used to ask Argos satellite passes and detailed content. It allows to return more data than the getXml command.

3.3.1. Parameters

Same parameters as getXml.

3.3.2. Return value

The return value is an octet-stream from which the data may be read. The data are compliant with the xml schema returned by getXsd.

The result is truncated to 10000 satellite passes. This limit may be changed by CLS.

3.4. getCsv

3.4.1. Parameters

Same parameters as getXml, plus the following parameters:

- showHeader
  Add a header line in the output, indicating the name of each column.

3.4.2. Return value

The return value is a string.

It contains lines of data.

Each line contains semicolon (;) separated values.

The result is truncated to 100 satellite passes. This limit may be changed by CLS.

3.5. getKml

3.5.1. Parameters

Same parameters as getXml,

- Minus the output content parameters (only locations are returned).
- Plus the following parameter:

- displayDescription
  At each placemark is associated a description box with mainly the platform location.
- displayDiagnostic
  The error ellipses associated to locations (class 0, 1, 2, 3 only) are returned in a folder, distinct from the location folder.

The result is truncated to 1000 satellite passes. This limit may be changed by CLS.

3.5.2. Return value

The getKmlResponse is a string containing the data. The data are compliant with the kml 2.2 schema.

Only passages with locations are returned.

The details of the kml elements is given in chapter 4.2

The result is truncated to 1000 satellite passes. This limit may be changed by CLS.

3.6. getObsXml

This request is used to ask Argos platform observations in xml format.

3.6.1. Parameters

Authentication: required

- username
  authentication user name. Same as for present customer ArgosWeb and Telnet access.
- password
  authentication password. Same as for present customer ArgosWeb and Telnet access.

Required data selection: choice between

- platformId
  comma separated list of platforms.
- programNumber
  comma separated list of programs. All platforms of the program will be accessed.
- wmoNumber
  comma separated list of WMO numbers.

Output size: optional

- nbMaxObs
  maximum number of observations returned by request. If there are actually more observations, only the first ones, with respect to observation_date, will be returned to the user. By default, all observations will be returned.

Time selection: choice between

- startDate / endDate
  observation date begin of search and end of search. Only the 10 last days are available. If endDate is not specified, the current date is assumed.
- nbDaysFromNow
  returns only satellite passes from the last days. Only the 10 last days are available.

3.6.2. Return value

The getObsXmlResponse is a string containing the data.

3.7. getObsCsv

This request is used to ask Argos platform observations in comma-separated values format.

3.7.1. Parameters

Same as getObsXml request

3.7.2. Return value

The getObsXmlResponse is a string containing the data.

3.8. getPlatformList

This request is used to ask Argos platforms available for a user.

3.8.1. Parameters

Authentication: required

- username
  authentication user name. Same as for present customer ArgosWeb and Telnet access.
- password
  authentication password. Same as for present customer ArgosWeb and Telnet access.

3.8.2. Return value

Returns program and platform list with platform last information: last collect date, last location.

~~3.6.~~3.9. WSDL reference

The WSDL content below is for information only. The actual DWS WSDL file shall be downloaded at the URL defined in §2.3

```
<?xml version='1.0' encoding='UTF-8'?><wsdl:definitions name="DixService"
targetNamespace="http://service.dataxmldistribution.argos.cls.fr/"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:tns="http://service.dataxmldistribution.argos.cls.fr/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:x1="http://service.dataxmldistribution.argos.cls.fr/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <wsdl:types>
<xs:schema elementFormDefault="qualified"
targetNamespace="http://service.dataxmldistribution.argos.cls.fr/types"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:tns="http://service.dataxmldistribution.argos.cls.fr/types"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:x1="http://service.dataxmldistribution.argos.cls.fr/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

   <xs:complexType name="periodType">
    <xs:sequence>
     <xs:element name="startDate" type="xs:dateTime" />
     <xs:element minOccurs="0" name="endDate" type="xs:dateTime" />
    </xs:sequence>
   </xs:complexType>

   <xs:simpleType name="referenceDateType">
    <xs:restriction base="xs:string">
     <xs:enumeration value="BEST_MSG_DATE" />
     <xs:enumeration value="MODIFICATION_DATE" />
    </xs:restriction>
   </xs:simpleType>

   <xs:simpleType name="argDistribType">
    <xs:restriction base="xs:string">
     <xs:enumeration value="A" />
     <xs:enumeration value="O" />
     <xs:enumeration value="B" />
    </xs:restriction>
   </xs:simpleType>

   <xs:complexType name="baseRequestType">
    <xs:sequence>
     <xs:element name="username" type="xs:string" />
     <xs:element name="password" type="xs:string" />

     <xs:choice>
      <xs:element name="programNumber" type="xs:string" />
      <xs:element name="platformId" type="xs:string" />
     </xs:choice>
     <xs:element minOccurs="0" name="nbPassByPtt" type="xs:int" />

     <xs:choice>
      <xs:element name="period" type="tns:periodType" />
      <xs:element name="nbDaysFromNow" type="xs:int" />
     </xs:choice>
     <!-- By default the reference is BEST_MSG_DATE -->
     <xs:element minOccurs="0" name="referenceDate" type="tns:referenceDateType" />

     <xs:element minOccurs="0" name="locClass" type="xs:string" />
     <xs:element minOccurs="0" name="geographicArea" type="xs:string" />
     <xs:element minOccurs="0" name="compression" type="xs:int" />
     <xs:element minOccurs="0" name="mostRecentPassages" type="xs:boolean" />
    </xs:sequence>
   </xs:complexType>

   <xs:complexType name="xmlRequestType">
    <xs:complexContent>
     <xs:extension base="tns:baseRequestType">
      <xs:sequence>
       <xs:element minOccurs="0" name="displayLocation" type="xs:boolean" />
       <xs:element minOccurs="0" name="displayDiagnostic" type="xs:boolean" />
       <xs:element minOccurs="0" name="displayMessage" type="xs:boolean" />
       <xs:element minOccurs="0" name="displayCollect" type="xs:boolean" />
       <xs:element minOccurs="0" name="displayRawData" type="xs:boolean" />
       <xs:element minOccurs="0" name="displaySensor" type="xs:boolean" />
       <xs:element minOccurs="0" name="argDistrib" type="tns:argDistribType" />
       <xs:element minOccurs="0" name="displayImageLocation" type="xs:boolean" />
       <xs:element minOccurs="0" name="displayHexId" type="xs:boolean" />
      </xs:sequence>
     </xs:extension>
    </xs:complexContent>
   </xs:complexType>


   <xs:complexType name="csvRequestType">
    <xs:complexContent>
     <xs:extension base="tns:xmlRequestType">
      <xs:sequence>
       <xs:element minOccurs="0" name="showHeader" type="xs:boolean" />
      </xs:sequence>
     </xs:extension>
    </xs:complexContent>
   </xs:complexType>

   <xs:complexType name="kmlRequestType">
    <xs:complexContent>
     <xs:extension base="tns:baseRequestType">
      <xs:sequence>
       <xs:element minOccurs="0" name="displayDescription" type="xs:boolean" />
       <xs:element minOccurs="0" name="displayDiagnostic" type="xs:boolean" />
      </xs:sequence>
     </xs:extension>
    </xs:complexContent>
   </xs:complexType>

   <xs:complexType name="xsdRequestType">
    <xs:sequence />
   </xs:complexType>

   <xs:complexType name="stringResponseType">
    <xs:sequence>
     <xs:element minOccurs="0" name="return" type="xs:string" />
    </xs:sequence>
   </xs:complexType>

   <xs:complexType name="streamResponseType">
    <xs:sequence>
     <xs:element minOccurs="0" name="return" ns1:expectedContentTypes="application/octet-
stream" type="xs:base64Binary" xmlns:ns1="http://www.w3.org/2005/05/xmlmime" />
    </xs:sequence>
   </xs:complexType>

   <xs:complexType name="DixException">
```

```
  <xs:sequence />
 </xs:complexType>

 <xs:element name="csvRequest" type="tns:csvRequestType" />
 <xs:element name="csvResponse" type="tns:stringResponseType" />

 <xs:element name="kmlRequest" type="tns:kmlRequestType" />
 <xs:element name="kmlResponse" type="tns:stringResponseType" />

 <xs:element name="streamXmlRequest" type="tns:xmlRequestType" />
 <xs:element name="streamXmlResponse" type="tns:streamResponseType" />

 <xs:element name="xmlRequest" type="tns:xmlRequestType" />
 <xs:element name="xmlResponse" type="tns:stringResponseType" />

 <xs:element name="xsdRequest" type="tns:xsdRequestType" />
 <xs:element name="xsdResponse" type="tns:stringResponseType" />

 <xs:element name="DixException" type="tns:DixException" />
</xs:schema>
</wsdl:types>
<wsdl:message name="streamXmlRequest">
  <wsdl:part element="x1:streamXmlRequest" name="params">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="xmlResponse">
  <wsdl:part element="x1:xmlResponse" name="params">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="csvRequest">
  <wsdl:part element="x1:csvRequest" name="params">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="xsdResponse">
  <wsdl:part element="x1:xsdResponse" name="params">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="kmlResponse">
  <wsdl:part element="x1:kmlResponse" name="params">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="kmlRequest">
  <wsdl:part element="x1:kmlRequest" name="params">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="xmlRequest">
  <wsdl:part element="x1:xmlRequest" name="params">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="csvResponse">
  <wsdl:part element="x1:csvResponse" name="params">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="streamXmlResponse">
  <wsdl:part element="x1:streamXmlResponse" name="params">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="DixException">
  <wsdl:part element="x1:DixException" name="DixException">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="xsdRequest">
  <wsdl:part element="x1:xsdRequest" name="params">
  </wsdl:part>
</wsdl:message>
<wsdl:portType name="DixServicePortType">
  <wsdl:operation name="getCsv">
    <wsdl:input message="tns:csvRequest">
  </wsdl:input>
    <wsdl:output message="tns:csvResponse">
  </wsdl:output>
    <wsdl:fault message="tns:DixException" name="DixException">
  </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="getStreamXml">
```

```
      <wsdl:input message="tns:streamXmlRequest">
    </wsdl:input>
      <wsdl:output message="tns:streamXmlResponse">
    </wsdl:output>
      <wsdl:fault message="tns:DixException" name="DixException">
    </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="getKml">
      <wsdl:input message="tns:kmlRequest">
    </wsdl:input>
      <wsdl:output message="tns:kmlResponse">
    </wsdl:output>
      <wsdl:fault message="tns:DixException" name="DixException">
    </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="getXml">
      <wsdl:input message="tns:xmlRequest">
    </wsdl:input>
      <wsdl:output message="tns:xmlResponse">
    </wsdl:output>
      <wsdl:fault message="tns:DixException" name="DixException">
    </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="getXsd">
      <wsdl:input message="tns:xsdRequest">
    </wsdl:input>
      <wsdl:output message="tns:xsdResponse">
    </wsdl:output>
      <wsdl:fault message="tns:DixException" name="DixException">
    </wsdl:fault>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="DixServiceSoapBinding" type="tns:DixServicePortType">
    <soap12:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="getCsv">
      <soap12:operation soapAction="Get Argos data on CSV format" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
      <wsdl:fault name="DixException">
        <soap12:fault name="DixException" use="literal" />
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="getStreamXml">
      <soap12:operation soapAction="" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
      <wsdl:fault name="DixException">
        <soap12:fault name="DixException" use="literal" />
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="getKml">
      <soap12:operation soapAction="Get Argos data on KML format" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
      <wsdl:fault name="DixException">
        <soap12:fault name="DixException" use="literal" />
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="getXml">
      <soap12:operation soapAction="Get Argos data on XML format" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
      <wsdl:fault name="DixException">
        <soap12:fault name="DixException" use="literal" />
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="getXsd">
      <soap12:operation soapAction="Get the XSD for the XML format" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
      <wsdl:fault name="DixException">
        <soap12:fault name="DixException" use="literal" />
      </wsdl:fault>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="DixService">
    <wsdl:port binding="tns:DixServiceSoapBinding" name="DixServicePort">
      <soap12:address location="http://argos-qt.cls.fr:8989/argosDdd/services/DddService" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

4. Description of the received data

4.1. Parameter description for getXml and getStreamXml

Sequence/Element	Description	Unit
data		
| program		
| programNumber	Argos program number	
| platform		
| platformId	Identification number of the platform.	
| platformType	Platform type (activity code)	
| platformName	Platform name	
| platformModel	Platform model	
| platformHexId	Platform hexadecimal identifier, i.e: primary_hcode || secondary_hcode	
| satellitePass		
| satellite	Satellite name (NK, NP, MA, …)	
| bestMsgDate	Satellite best message date/time	
| duration	Pass duration	s
| nbMessage	Number of Argos messages received	
| message120	Number of messages > -120 dB	
| bestLevel	dB Level of best message	dBm
| frequency	Measured Frequency of Argos passage	Hz

| location		
| locationDate	Date/time of localization	
| latitude	Latitude of platform	-90..90°
| longitude	Longitude of platform	-180..180°
| altitude	Altitude of platform (from terrain model or from declaration or fixed to 0)	m
| locationClass	Location quality class (Z,B,A,0,1,2,3,G)	
| gpsSpeed	Instantaneous speed of GPS receiver	m/s
| gpsHeading	Instantaneous heading of GPS receiver, 0°=North, clockwise.	0..360°
| diagnostic		
| latitude2	Image latitude of platform	-90..90°
| longitude2	Image longitude of platform	-180..180°
| altitude2	Image altitude of platform	m
| index	Location quality index (0..99)	
| nopc	Number of likely checks	
| errorRadius	Error circle radius (same circle surface as the ellipse)	m
| semiMajor	Ellipse semi-major axis	m
| semiMinor	Ellipse semi-minor axis	m
| orientation	Ellipse orientation of major axis, 0°=North, clockwise.	0..180°
| hdop	Geometric dillution of precision	m/Hz
| message		
| bestDate	Collect date chosen after compression	
| compression	Number of identical Argos messages	
| collect		
| type	Message type: L=Low Rate, H=High Rate	
| alarm	Message alarm level : N=None, D=Detection, C=Confirmation, …	
| concatenated	Indicates if the message received is issued from concatenation (Y/N)	
| date	Date/Time of collection	
| level	Measured message level	dBm
| doppler	Measured Doppler of Argos message	Hz
| rawData	Argos message in hexadecimal format.	

| format		
| formatOrder	Format order number	
| formatName	Format name	
| sensor		
| order	Sensor order number	
| name	Name of the sensor	
| valueType	Type of the sensor value: I=Integer, R=real, S=String, L=location, D=Date/time	
| value	Value of numerical type sensors (I, R)	
| valueStr	Value of string sensors (S)	
| errors		
| error		

Table 2: parameter meanings and units

4.2. XML Schema for getXml and getStreamXml

This xml schema is for information only. The applicable xml schema should be accessed with the getXml function.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <!-- W3C specifications   http://www.w3.org/TR/xmlschema-0  -->
  <!-- DATA DECLARATION -->
  <xs:complexType name="data">
    <xs:sequence>
      <xs:element name="program" type="program" minOccurs="0" maxOccurs="unbounded" />
      <xs:element name="errors" type="errors" minOccurs="0" />
    </xs:sequence>
    <xs:attribute name="version" type="xs:string" />
  </xs:complexType>
  <!-- PROGRAM DECLARATION -->
  <xs:complexType name="program">
    <xs:sequence>
      <xs:element name="programNumber" type="xs:int" minOccurs="0" />
      <xs:element name="platform" type="platform" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
  <!-- PLATFORM DECLARATION -->
  <xs:complexType name="platform">
    <xs:sequence>
      <xs:element name="platformId" type="xs:int" minOccurs="0" />
      <xs:element name="platformType" type="xs:string" minOccurs="0" />
      <xs:element name="platformName" type="xs:string" minOccurs="0" />
      <xs:element name="platformModel" type="xs:string" minOccurs="0" />
      <xs:element name="platformHexId" type="xs:string" minOccurs="0" />
      <xs:element name="satellitePass" type="satellitePass" minOccurs="0"
maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
  <!-- SATELLITE PASS DECLARATION -->
  <xs:complexType name="satellitePass">
    <xs:sequence>
      <xs:element name="satellite" type="xs:string" minOccurs="0" />
      <xs:element name="bestMsgDate" type="xs:dateTime" minOccurs="0"/>
      <!-- unit : second -->
      <xs:element name="duration" type="xs:int" minOccurs="0" />
      <xs:element name="nbMessage" type="xs:int" minOccurs="0" />
      <xs:element name="message120" type="xs:int" minOccurs="0" />
      <!-- unit : dBm -->
      <xs:element name="bestLevel" type="xs:int" minOccurs="0" />
      <!-- unit : Hertz -->
      <xs:element name="frequency" type="xs:double" minOccurs="0" />
      <xs:element name="location" type="location" minOccurs="0" />
      <xs:element name="message" type="message" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
  <!-- LOCATION DECLARATION -->
  <xs:complexType name="location">
    <xs:sequence>
      <!-- GMT -->
      <xs:element name="locationDate" type="xs:dateTime" minOccurs="0" />
      <!-- range : -90°..+90° -->
      <xs:element name="latitude" type="xs:double" minOccurs="0" />
      <!-- range : -180°..+180° -->
      <xs:element name="longitude" type="xs:double" minOccurs="0" />
      <!-- unit : meter -->
      <xs:element name="altitude" type="xs:double" minOccurs="0" />
      <xs:element name="locationClass" type="xs:string" minOccurs="0" />
      <!-- unit : meter/second -->
      <xs:element name="gpsSpeed" type="xs:float" minOccurs="0" />
      <!-- range : 0°..+360°, clockwise -->
      <xs:element name="gpsHeading" type="xs:float" minOccurs="0" />
      <xs:element name="diagnostic" type="diagnostic" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
  <!-- DIAGNOSTIC DECLARATION -->
  <xs:complexType name="diagnostic">
    <xs:sequence>
      <!-- range : -90°..+90° -->
      <xs:element name="latitude2" type="xs:float" minOccurs="0" />
      <!-- range : -180°..+180° -->
      <xs:element name="longitude2" type="xs:float" minOccurs="0" />
      <!-- unit : meter -->
      <xs:element name="altitude2" type="xs:float" minOccurs="0" />
      <xs:element name="index" type="xs:int" minOccurs="0" />
      <xs:element name="nopc" type="xs:int" minOccurs="0" />
      <!-- unit : meter -->
      <xs:element name="errorRadius" type="xs:float" minOccurs="0" />
      <!-- unit : meter -->
      <xs:element name="semiMajor" type="xs:float" minOccurs="0" />
      <!-- unit : meter -->
      <xs:element name="semiMinor" type="xs:float" minOccurs="0" />
      <!-- range 0°..180°, clockwise -->
      <xs:element name="orientation" type="xs:float" minOccurs="0" />
      <xs:element name="hdop" type="xs:string" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
  <!-- MESSAGE DECLARATION -->
  <xs:complexType name="message">
    <xs:sequence>
      <!-- GMT -->
      <xs:element name="bestDate" type="xs:dateTime" minOccurs="0" />
      <xs:element name="compression" type="xs:int" minOccurs="0" />
      <xs:element name="collect" type="collect" minOccurs="0" maxOccurs="unbounded" />
      <xs:element name="format" type="format" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
  <!-- COLLECT DECLARATION -->
  <xs:complexType name="collect">
    <xs:sequence>
      <xs:element name="type" type="xs:string" minOccurs="0" />
      <xs:element name="alarm" type="xs:string" minOccurs="0" />
      <xs:element name="concatenated" type="xs:string" minOccurs="0" />
      <!-- GMT -->
      <xs:element name="date" type="xs:dateTime" minOccurs="0" />
      <!-- unit : dBm -->
      <xs:element name="level" type="xs:float" minOccurs="0" />
      <!-- unit : Hertz -->
      <xs:element name="doppler" type="xs:double" minOccurs="0" />
      <xs:element name="rawData" type="xs:string" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
  <!-- FORMAT DECLARATION -->
  <xs:complexType name="format">
    <xs:sequence>
      <xs:element name="formatOrder" type="xs:int" minOccurs="0" />
      <xs:element name="formatName" type="xs:string" minOccurs="0" />
      <xs:element name="sensor" type="sensor" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
  <!-- SENSOR DECLARATION -->
  <xs:complexType name="sensor">
    <xs:sequence>
      <xs:element name="order" type="xs:int" minOccurs="0" />
      <xs:element name="name" type="xs:string" minOccurs="0" />
      <xs:element name="valueType" type="xs:string" minOccurs="0" />
      <xs:element name="value" type="xs:string" minOccurs="0" />
      <xs:element name="valueStr" type="xs:string" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
  <!-- ERRORS DECLARATION -->
  <xs:complexType name="errors">
    <xs:sequence>
      <xs:element name="error" type="xs:string" minOccurs="0" maxOccurs="unbounded">
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

4.3. KML Schema for getKml

The kml schema is compliant with the KML 2.2 specification described at http://schemas.opengis.net/kml/.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<kml xmlns="http://www.opengis.net/kml/2.2" xmlns:gx="http://www.google.com/kml/ext/2.2">
  <Document>
    <name>CLS - [creation date]</name>
    <LookAt>
      <gx:TimeStamp>
        <when>[date]</when>
      </gx:TimeStamp>
      <range></range>
    </LookAt>
    <Style id="POINT">[style definition]</Style>
    <Style id="LAST_POINT">[style definition]</Style>
    <Style id="TRAJECTORY">[style definition]</Style>
    <Style id="ERROR_ELLIPSE">[style definition]</Style>
    <Folder>
      <name>Program [program number]</name>
      <Folder>
        <name>Platform [platform id]</name>
        <Placemark>
          <name>[platform id]</name>
          <TimeSpan>
            <begin>[location date]</begin>
          </TimeSpan>
          <styleUrl>#LAST_POINT</styleUrl>
          <Point>
            <coordinates>[longitude],[latitude]</coordinates>
          </Point>
        </Placemark>
        <Placemark>
          <name>Trajectory</name>
          <visibility>1</visibility>
          <open>0</open>
          <styleUrl>#TRAJECTORY</styleUrl>
          <LineString>
            <extrude>0</extrude>
            <tessellate>1</tessellate>
            <coordinates>[longitude],[latitude] ... [longitude],[latitude]</coordinates>
          </LineString>
        </Placemark>
        <Folder>
          <name>Locations</name>
          <Placemark>
            <TimeSpan>
              <begin>[location date]</begin>
            </TimeSpan>
            <styleUrl>#POINT</styleUrl>
            <Point>
              <coordinates>[longitude],[latitude]</coordinates>
            </Point>
          </Placemark>
          ... (other downloaded locations) ...
        </Folder>
        <Folder>
          <name>Error ellipse</name>
          <Placemark>
            <styleUrl>#ERROR_ELLIPSE</styleUrl>
            <LinearRing>
              <coordinates>[[longitude],[latitude],0.0]*100</coordinates>
            </ LinearRing >
          </Placemark>
          ... (other downloaded error ellipses) ...
        </Folder>
      </Folder>
      ...(other downloaded platforms) ...
    </Folder>
    ...(other downloaded programs) ...
    [<ExtendedData>
      <Data name="error code">
        <displayName>[error description]</displayName>
        <value>[error code]</value>
      </Data>
    </ExtendedData>]
  </Document>
</kml>
```

4.4. CSV Schema for getCsv

For location request, the returned columns are:

```
“programNumber”;”platformId”;”platformType”;”platformModel”;”platformName”;”satellite”;”duration”;”nbMessage”;”message120”;”bestLevel”;”locationDate”;”latitude”;”longitude”;”altitude”;”locationClass”;”gpsSpeed”;”gpsHeading”
```

There is one line per satellite pass

For diagnostic request, the returned columns are:

“programNumber”;”platformId”;”platformType”;”platformModel”;”platformName”;”satellite”;”duration”;”nbMessage”;”message120”;”bestLevel”;”locationDate”;”latitude”;”longitude”;”altitude”;”locationClass”;”gpsSpeed”;”gpsHeading”;”index”;”nopc”;”errorRadius”;”semiMajor”;”semiMinor”;”orientation”;”hdop”

There is one line per satellite pass

For message request, the returned columns are:

“programNumber”;”platformId”;”platformType”;”platformModel”;”platformName”;”satellite”;”duration”;”nbMessage”;”message120”;”bestLevel”;”bestDate”;”compression”

There is one line per satellite pass and compressed message (different message content)

For collect request, the returned columns are:

“programNumber”;”platformId”;”platformType”;”platformModel”;”platformName”;”satellite”;”duration”;”nbMessage”;”message120”;”bestLevel”;”bestDate”;”compression”;”type”;”alarm”;”concatenated”;”date”;”frequency”;”level”

There is one line per received message

For rawdata request, the returned columns are:

“programNumber”;”platformId”;”platformType”;”platformModel”;”platformName”;”satellite”;”duration”;”nbMessage”;”message120”;”bestLevel”;”bestDate”;”compression”;”type”;”alarm”;”concatenated”;”date”;”frequency”;”level”;”rawData”

There is one line per received message

For sensor request, the returned columns are:

“programNumber”;”platformId”;”platformType”;”platformModel”;”platformName”;”satellite”;”duration”;”nbMessage”;”message120”;”bestLevel”;”bestDate”;”compression”;”formatOrder”;”formatName”;[“order”;”name”;”valueType”;”value”;”valueStr”;]…

There is one line per satellite pass and compressed message (different message content) and per format. The number of columns is adjusted to the number of sensors by message. Each sensor adds the following columns [“order”;”name”;”valueType”;”value”;”valueStr”;]

With displayHexId, the returned columns are:

“programNumber”;”platformId”;”platformType”;”platformModel”;”platformName”;”platformHexId”;”satellite”;…

4.5. Error codes

```
<error code="1">missing url parameters</error>
```

A parameter error has not been detected at SOAP level

```
<error code="2">max response reached</error>
```

The satellite pass limit has been reached

```
<error code="3">authentification error</error>
```

The username/password authentification is invalid

```
<error code="4">no data</error>
```

No data corresponding to the request could be found

```
<error code="9">start date upper than end date</error>
```

The startDate/endDate is invalid

4.6. Parameter description for getObsXml

Sequence/Element	Description	Unit
DIST		
⌊ PTT		
⌊ id	Identification number of the platform.	
⌊ ptt_type	Platform type (activity code)	
⌊ ptt_model	Platform model	
⌊ wmo	WMO number	
⌊ program	Argos program number	
⌊ OBS	Observation data	

⌊ id	Internal identifier.	
⌊ obs_date	Observation date	
⌊ loc_date	Location date of the observation	
⌊ latitude	Latitude of platform	-90..90°
⌊ longitude	Platform hexadecimal identifier, i.e: primary_hcode || secondary_hcode	-180..180°
⌊ loc_class	Location quality class (Z,B,A,0,1,2,3,G)	
⌊ LEVEL		m
⌊ value	Sensor height (>0) or depth (<0)	
⌊ component	Sensors at this level	
⌊ element_name	Sensor name	
⌊ value	Sensor value	-90..90°

Table 3: parameter meanings and units

4.7. CSV Schema for getObsCsv

For Observation request, the returned columns are:

```
observationId;platformId;wmo;observationDate;latitude;longitude;locationDate;locationQualityId;level;<sensor list>
```

There is one line per observation and per level

4.8. Parameter description for getPlatformList

Sequence/Element	Description	Unit
data		
⌊ program		
⌊ programNumber	Argos program number	
⌊ platform		
⌊ platformId	Identification number of the platform.	
⌊ lastCollectDate	Most recent satellite collection date	
⌊ lastLocationDate	Most recent platform location date	
⌊ lastLatitude	Most recent platform location	-90..90°
⌊ lastLongitude	Most recent platform location	-180..180°

Table 4: parameter meanings and units

5. Specification Maintenance

CLS regularly maintains the specification to meet new requirements and to incorporate practical experience gathered by using the web service. This concerns the WSDL file, the xsd file, and the kml contents.

As much as possible, changes are done to ensure backward compatibility, but that may not be always possible.

To ensure backward compatibility, clients shall be prepared to accept an unknown response and treat this situation as an indication that the XML schema version has changed.

In the following tables, the Operation column may be one of “Change”, “Add”, and “Delete”. “Delete” operation of non-mandatory xml elements and “Add” operations shall pose no backward compatibility issue.

5.1. Release notes

5.1.1. WSDL

Version	Operation	Change	Reason
1.0			Initial revision
1.1	Add	GetKml request and response types.	The interface is now finalized.
1.2	Add	<displayHexId> Boolean option in getXml, getStreamXml, getCsv requests.	Hexadecimal ID is used in coding of PMT FCS, it is needed in the distributed data to verify FCS.
1.2	Add	<DisplayDiagnostic> Boolean option in getKml request.	Location error ellipses will appear in the kml distributed data.
1.3	Add	<referenceDate> Enum(BEST_MSG_DATE, MODIFICATION_DATE) option in all getXxx requests	Requests use best message date to extract passages. But passages are not processed by date order by the CTA. With this option, the user may retrieve the most recently processed passages.
1.4	Add	Service to access the customer’s platform list and last locations	Request for change
1.4	Add	Service to access platform observations	Request for change

Table 5: wsdl release notes

XML schema (xsd)

Version	Operation	Change	Reason
1.0			Initial revision
1.1	Add	A <frequency> element has been added to <satellitePass> sequence. Its type is “double”.	Frequency is the platform estimated frequency during the satellite pass.

1.1	Delete	The <frequency> element has been removed from <collect> sequence.	Same as above.
1.2	Add	<platformHexId> will be returned in the <platform> sequence if the request flag <displayHexId> is true.	
1.3	Add	<bestMsgDate> added in <satellitePass> sequence	Anomaly fix in GPS location distribution.

Table 6: xml schema release notes

5.2. Planned changes

This chapter describes issues on the interface itself, not on the underlying system.

These issues are expected to be corrected in new interface document releases, and a corresponding server application upgrade. The service users will be informed of the expected deployment date. The server at CLS will support only the latest version.

5.2.1. WSDL

Version	Operation	Change	Reason
		No change planned	
~~TBD~~	~~Add~~	~~Service to access the customer's platform list and last locations~~	~~Request for change~~
~~TBD~~	~~Add~~	~~Service to access platform observations~~	~~Request for change~~

Table 7: wsdl planned changes

5.2.2. XML schema (xsd)

Version	Operation	Change	Reason
		No change planned	

Table 8: xml schema planned changes

6. Examples

6.1. getXml service

6.1.1. SOAP request

```
<soap:Envelope                              xmlns:soap=”http://www.w3.org/2003/05/soap-envelope”
xmlns:typ=”http://service.dataxmldistribution.argos.cls.fr/types”>
  <soap:Header/>
  <soap:Body>
    <typ:xmlRequest>
      <typ:username>mturiot</typ:username>
      <typ:password>qt</typ:password>
      <typ:platformId>1</typ:platformId>
      <typ:nbPassByPtt>2</typ:nbPassByPtt>
      <typ:nbDaysFromNow>10</typ:nbDaysFromNow>
      <typ:mostRecentPassages>true</typ:mostRecentPassages>
    </typ:xmlRequest>
  </soap:Body>
</soap:Envelope>
```

6.1.2. SOAP response

```
<soap:Envelope xmlns:soap=”http://www.w3.org/2003/05/soap-envelope”>
  <soap:Body>
    <xmlResponse xmlns=”http://service.dataxmldistribution.argos.cls.fr/types”>
      <return><![CDATA[<?xml version=”1.0” encoding=”ISO-8859-1”?><data
version=”1.0”><program><programNumber>1</programNumber><platform><platformId>1</platformId
><platformType>ORBITO</platformType><platformName>TOULOUSE</platformName><satellitePass><
satellite>NP</satellite><duration>900</duration><nbMessage>26</nbMessage><message120>21</mess
age120><bestLevel>-111</bestLevel><location><locationDate>2010-05-
12T13:25:46.000Z</locationDate><latitude>43.54999</latitude><longitude>1.4853200000000015</lon
gitude><altitude>211.0</altitude><locationClass>3</locationClass><gpsSpeed>0.0</gpsSpeed><gpsHe
ading>0.0</gpsHeading></location></satellitePass><satellitePass><satellite>NN</satellite><duration>
869</duration><nbMessage>26</nbMessage><message120>19</message120><bestLevel>-
111</bestLevel><location><locationDate>2010-05-
12T13:53:46.000Z</locationDate><latitude>43.54984</latitude><longitude>1.486060000000009</longi
tude><altitude>211.0</altitude><locationClass>2</locationClass></location></satellitePass></platfor
m></program></data>]]></return>
    </xmlResponse>
  </soap:Body>
</soap:Envelope>
```

6.2. getStreamXml service

6.2.1. SOAP request

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:typ="http://service.dataxmldistribution.argos.cls.fr/types">
  <soap:Header/>
  <soap:Body>
    <typ:streamXmlRequest>
      <typ:username>mturiot</typ:username>
      <typ:password>qt</typ:password>
      <typ:platformId>1</typ:platformId>
      <!--Optional:-->
      <typ:nbPassByPtt>1</typ:nbPassByPtt>
      <typ:nbDaysFromNow>2</typ:nbDaysFromNow>
    </typ:streamXmlRequest>
  </soap:Body>
</soap:Envelope>
```

6.2.2. SOAP response

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
    <streamXmlResponse xmlns="http://service.dataxmldistribution.argos.cls.fr/types">
      <return>
        <xop:Include href="cid:eb697d96-5ed7-4887-863e-eb1fab807803-
829@service.dataxmldistribution.argos.cls.fr"
xmlns:xop="http://www.w3.org/2004/08/xop/include"/>
      </return>
      </streamXmlResponse>
   </soap:Body>
</soap:Envelope>
```

For information on XOP, see http://www.w3.org/TR/xop10/.

See also http://en.wikipedia.org/wiki/XML-binary_Optimized_Packaging

6.3. getCsv service

6.3.1. SOAP request

```
<soap:Envelope xmlns:soap=”http://www.w3.org/2003/05/soap-envelope”
xmlns:typ=”http://service.dataxmldistribution.argos.cls.fr/types”>
   <soap:Header/>
   <soap:Body>
      <typ:csvRequest>
         <typ:username>mturiot</typ:username>
         <typ:password>qt</typ:password>
         <!—You have a CHOICE of the next 2 items at this level→
         <typ:platformId>1</typ:platformId>
         <!—Optional:→
         <typ:nbPassByPtt>10</typ:nbPassByPtt>
         <!—You have a CHOICE of the next 2 items at this level→
         <typ:nbDaysFromNow>2</typ:nbDaysFromNow>
         <typ:displayLocation>true</typ:displayLocation>
         <typ:showHeader>true</typ:showHeader>
      </typ:csvRequest>
   </soap:Body>
</soap:Envelope>
```

6.3.2. SOAP response

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
   <soap:Body>
      <csvResponse xmlns="http://service.dataxmldistribution.argos.cls.fr/types">
         <return>"1";"1";"ORBITO";"";"TOULOUSE";"NL";"690";"21";"13";"-114";"2010-05-
18T08:38:16.000Z";"43.5513";"1.4849399999999946";"211.0";"3";"";"";
"1";"1";"ORBITO";"";"TOULOUSE";"NM";"869";"27";"23";"-111";"2010-05-
18T08:58:01.000Z";"43.54949";"1.4836199999999735";"211.0";"3";"";"";
"1";"1";"ORBITO";"";"TOULOUSE";"NM";"720";"23";"16";"-112";"2010-05-
18T10:37:46.000Z";"43.5506";"1.4856700000000274";"211.0";"2";"";"";
"1";"1";"ORBITO";"";"TOULOUSE";"NP";"360";"13";"1";"-119";"2010-05-
18T10:45:01.000Z";"43.55341";"1.4783100000000218";"211.0";"1";"";"";
"1";"1";"ORBITO";"";"TOULOUSE";"NN";"420";"11";"6";"-118";"2010-05-
18T11:11:01.000Z";"43.55073";"1.4876100000000179";"211.0";"2";"";"";
"1";"1";"ORBITO";"";"TOULOUSE";"NM";"59";"2";"0";"-130";"2010-05-
18T12:17:01.000Z";"43.55179";"1.4826199999999972";"211.0";"B";"";"";
"1";"1";"ORBITO";"";"TOULOUSE";"NP";"900";"31";"24";"-112";"2010-05-
18T12:23:46.000Z";"43.54923";"1.4842699999999809";"211.0";"2";"";"";
"1";"1";"ORBITO";"";"TOULOUSE";"NN";"930";"31";"23";"-109";"2010-05-
18T12:50:31.000Z";"43.54979";"1.484199999999987";"211.0";"2";"";"";
"1";"1";"ORBITO";"";"TOULOUSE";"NP";"840";"26";"16";"-113";"2010-05-
18T14:04:46.000Z";"43.54973";"1.4854399999999828";"211.0";"3";"";"";
"1";"1";"ORBITO";"";"TOULOUSE";"NK";"660";"21";"15";"-113";"2010-05-
18T14:24:01.000Z";"43.55106";"1.4844600000000128";"211.0";"2";"";"";</return>
      </csvResponse>
   </soap:Body>
</soap:Envelope>
```

6.4. getKml service

This service is available.

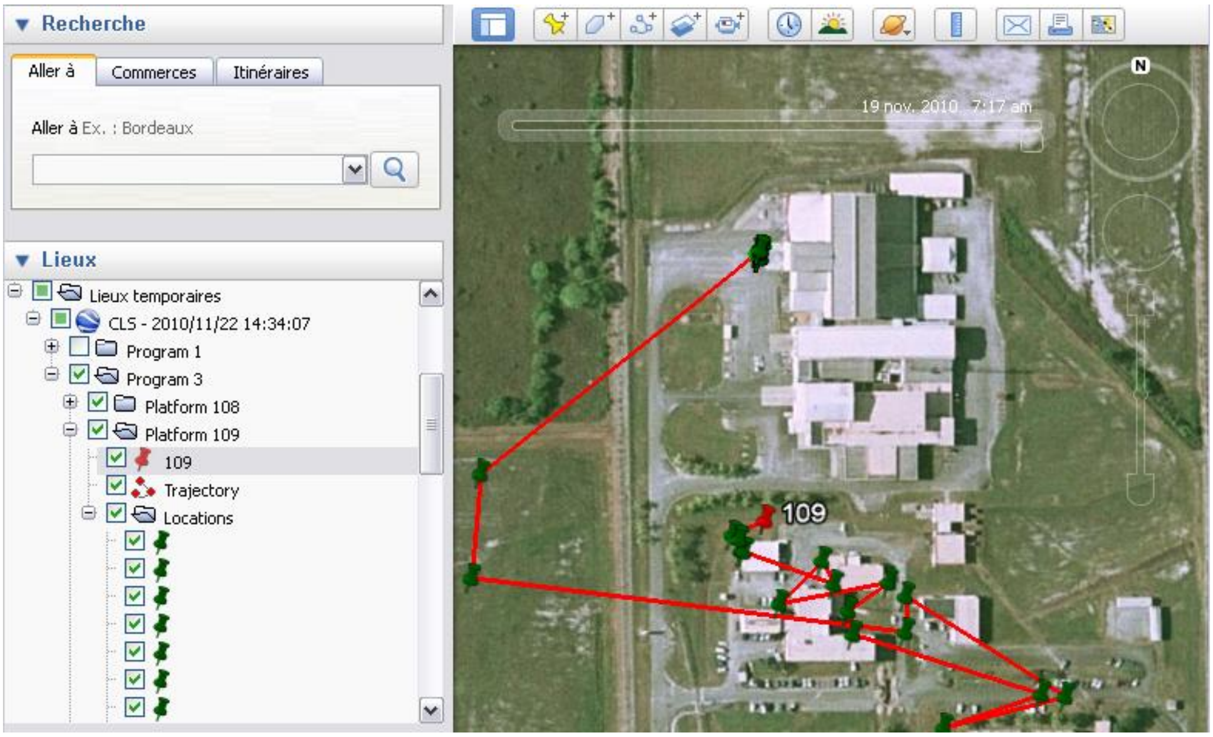
6.4.1. SOAP request

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:typ="http://service.dataxmldistribution.argos.cls.fr/types">
   <soap:Header/>
   <soap:Body>
      <typ:kmlRequest>
         <typ:username>mturiot</typ:username>
         <typ:password>qt</typ:password>
         <typ:platformId>1</typ:platformId>
         <typ:nbDaysFromNow>8</typ:nbDaysFromNow>
         <!--Optional:-->
         <typ:locClass>1,2,3</typ:locClass>
      </typ:kmlRequest>
   </soap:Body>
</soap:Envelope>
```

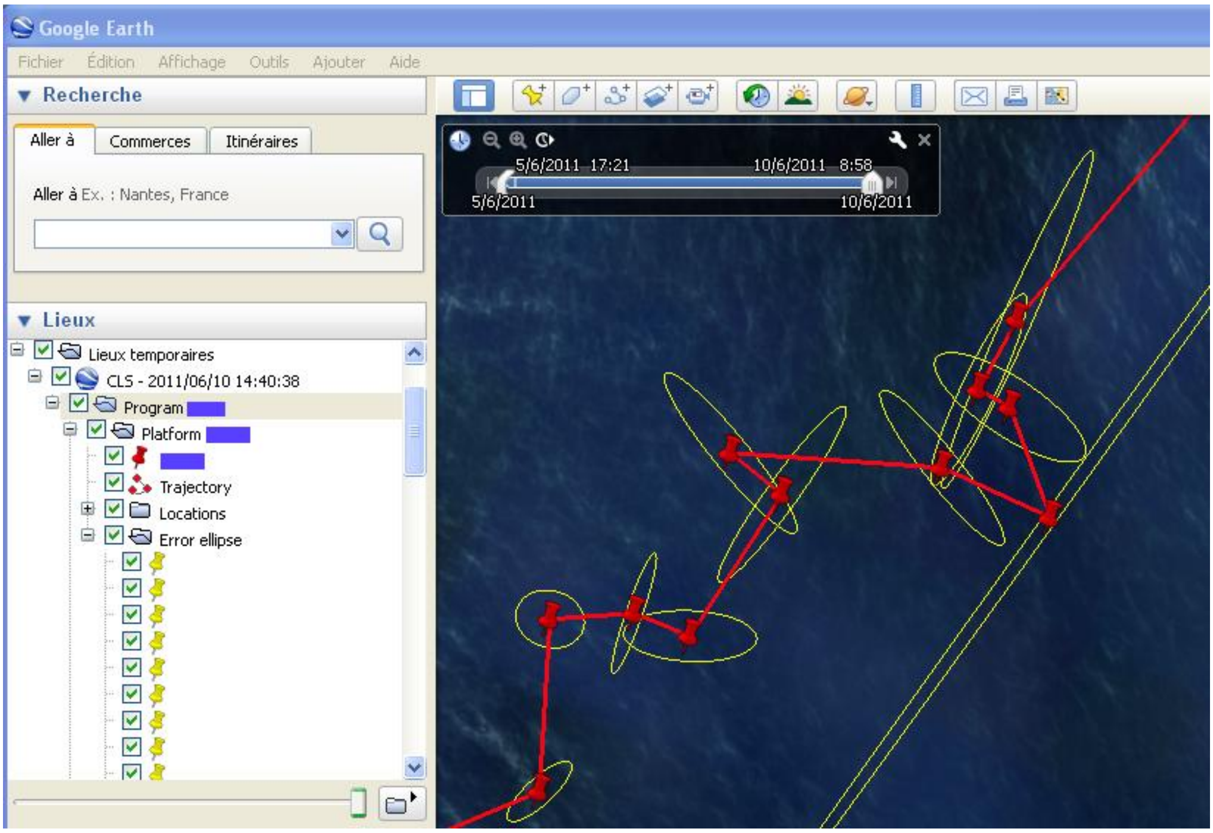
6.4.2. SOAP response

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
   <soap:Body>
      <kmlResponse xmlns="http://service.dataxmldistribution.argos.cls.fr/types">
         <return><![CDATA[<?xml version="1.0" encoding="ISO-8859-1"?><kml
xmlns="http://www.opengis.net/kml/2.2"
xmlns:gx="http://www.google.com/kml/ext/2.2"><Document><name>CLS - 2010/12/03
15:51:04</name><LookAt><gx:TimeStamp><when>2010-12-
03T15:51:04Z</when></gx:TimeStamp><range>9000000</range></LookAt><Style
id="POINT"><IconStyle><color>ff0000ff</color><scale>0.4</scale><Icon><href>http://maps
.google.com/mapfiles/kml/pushpin/red-
pushpin.png</href></Icon></IconStyle></Style><Style
id="LAST_POINT"><IconStyle><color>ff0000ff</color><scale>0.8</scale><Icon><href>http:/
/maps.google.com/mapfiles/kml/pushpin/red-
pushpin.png</href></Icon></IconStyle></Style><Style
id="TRAJECTORY"><LineStyle><color>ff0000ff</color><width>2.0</width></LineStyle></Styl
e><Folder><name>Program 1</name><Folder><name>Platform
1</name><Placemark><name>1</name><TimeSpan><begin>2010-12-
03T14:57:56Z</begin></TimeSpan><styleUrl>#LAST_POINT</styleUrl><Point><coordinates>1.4
8507,43.55014</coordinates></Point></Placemark><Placemark><name>Trajectory</name><visi
bility>1</visibility><open>0</open><styleUrl>#TRAJECTORY</styleUrl><LineString><extrud
e>0</extrude><tessellate>1</tessellate><coordinates>1.48508,43.55015 1.48508,43.55015
1.48507,43.55015 1.48507,43.55015 1.48507,43.55015 1.48508,43.55015 1.48507,43.55015
1.48507,43.55015 1.48507,43.55015 1.48506,43.55015 1.48506,43.55015 1.48506,43.55015
1.48507,43.55015 1.48506,43.55015 1.48507,43.55015 1.48508,43.55014 1.48508,43.55014
1.48508,43.55014
1.48507,43.55014</coordinates></LineString></Placemark><Folder><name>Locations</name><
Placemark><TimeSpan><begin>2010-12-
03T04:57:02Z</begin></TimeSpan><styleUrl>#POINT</styleUrl><Point><coordinates>1.48508,
43.55015</coordinates></Point></Placemark><Placemark><TimeSpan><begin>2010-12-
03T06:11:34Z</begin></TimeSpan><styleUrl>#POINT</styleUrl><Point><coordinates>1.48508,
43.55015</coordinates></Point></Placemark><Placemark><TimeSpan><begin>2010-12-
03T06:35:58Z</begin></TimeSpan><styleUrl>#POINT</styleUrl><Point><coordinates>1.48507,
43.55015</coordinates></Point></Placemark><Placemark><TimeSpan><begin>2010-12-
03T07:52:21Z</begin></TimeSpan><styleUrl>#POINT</styleUrl><Point><coordinates>1.48507,
43.55015</coordinates></Point></Placemark><Placemark><TimeSpan><begin>2010-12-
03T08:07:26Z</begin></TimeSpan><styleUrl>#POINT</styleUrl><Point><coordinates>1.48507,
43.55015</coordinates></Point></Placemark><Placemark><TimeSpan><begin>2010-12-
03T09:08:16Z</begin></TimeSpan><styleUrl>#POINT</styleUrl><Point><coordinates>1.48508,
43.55015</coordinates></Point></Placemark><Placemark><TimeSpan><begin>2010-12-
03T14:18:31Z</begin></TimeSpan><styleUrl>#POINT</styleUrl><Point><coordinates>1.48508,
43.55014</coordinates></Point></Placemark><Placemark><TimeSpan><begin>2010-12-
03T14:36:21Z</begin></TimeSpan><styleUrl>#POINT</styleUrl><Point><coordinates>1.48508,
43.55014</coordinates></Point></Placemark></Folder></Folder></Folder></Document></kml>
]]></return>
      </kmlResponse>
   </soap:Body>
</soap:Envelope>
```

6.4.3. Sample KML displayed in GoogleEarth



Example containing error ellipses:



Appendix A - List of acronyms

TBC	To be confirmed
TBD	To be defined
AD	Applicable Document
RD	Reference Document

APC	Argos Processing Center
CSV	Comma-Separated Values
DWS	Distribution Web Service
HTTP	Hypertext Transfer Protocol
KML	Keyhole Markup Language
MIME	Multipurpose Internet Mail Extensions
SOAP	Simple Object Access Protocol
WSDL	Web Services Description Language
XML	Extensible Markup Language
XOP	XML-binary Optimized Packaging