

CDPP - 3DVIEW

CDPP-IF-32600-534-GFI

Issue : 00 Date : 04/12/2015

Revision : 01 Date : 22/06/2016

MT : X Distribution Code : E

Ref.: CNES/CDPP-3DView/PRD/DOC/IF

**INTERFACES SPECIFICATIONS
3DVIEW**

Written by: BEIGBEDER Laurent CAUSSARIEU Stéphane	GFI INFORMATIQUE GFI INFORMATIQUE	Date:	
Approved by: TONIUTTI Jean-Philippe	GFI INFORMATIQUE	Date:	

INDEX SHEET

CONFIDENTIALITY:

NCKEYWORDS: **Web Services, Interface, Specification, CDPP, 3DView**

TITLE:

**Interfaces Specifications
3DView**

AUTHOR(S):

BEIGBEDER Laurent**GFI INFORMATIQUE****CAUSSARIEU Stéphane****GFI INFORMATIQUE**SUMMARY: **This document describes the CDPP 3DView software interfaces**RELATED DOCUMENTS: **Stand alone document.**

LOCALIZATION:

CNES/CDPP-3DView/PRD/DOCVOLUME: **1**

TOTAL NUMBER OF PAGES: **32**
INCLUDING PRELIMINARY PAGES: **6**
NUMBER OF SUPPL. PAGES: **0**

COMPOSITE DOCUMENT: **N**LANGUAGE: **EN**CONFIGURATION MANAGEMENT: **NG**

CM RESP.:

REASONS FOR EVOLUTION : **Update in conformance with the software version V1.11**CONTRACT: **Bon de commande MARCHE SOUS ACCORD-CADRE N° 151283**

HOST SYSTEM:

Microsoft Word 11.0 (11.0.5604)**L:\CLASSE1\Modèles word\GDOC V3.1.8\ModeleGDOCIndus_2015_en.dot****Version GDOC : v3.1.8**

INTERNAL DISTRIBUTION

Name	Entity	Internal Postal Box	Observations
DUFOURG Nicolas	DCT/ME/EU	612	
DURAND Joëlle	DCT/ME/EU	612	

EXTERNAL DISTRIBUTION

Name	Entity	Observations
BEIGBEDER Laurent	GFI INFORMATIQUE	
CAUSSARIEU Stéphane	GFI INFORMATIQUE	
POPESCU Daniel	GFI INFORMATIQUE	
TONIUTTI Jean-Philippe	GFI INFORMATIQUE	

CHANGES

Issue	Rev.	Date	Reference, Author(s), Reasons for evolution
00	01	22/06/2016	CNES/CDPP-3DView/PRD/DOC/IF BEIGBEDER Laurent GFI INFORMATIQUE CAUSSARIEU Stéphane GFI INFORMATIQUE Update in conformance with the software version V1.11
00	00	04/12/2015	CNES/CDPP-3DView/PRD/DOC/IF BEIGBEDER Laurent GFI INFORMATIQUE Creation of the document

TABLE OF CONTENTS

GLOSSARY AND LIST OF TBC AND TBD ITEMS.....	1
1. OVERVIEW.....	2
1.1. APPLICABLE DOCUMENTS	2
1.2. REFERENCE DOCUMENTS.....	2
2. INTRODUCTION.....	3
3. 3DVIEW FUNCTIONNAL WEB SERVICES.....	4
3.1. LISTFRAMES.....	4
3.2. LISTFRAMES2	4
3.3. LISTBODIES	4
3.4. LISTBODIESINREGION	5
3.5. LISTBOIS	5
3.6. LISTFILES.....	5
3.7. GETCOVERAGE	5
3.8. GETSIZE	6
3.9. LISTSTARSUBSETS.....	6
3.10. LISTSTARS	6
3.11. LISTINSTS	6
3.12. LISTGS	6
3.13. LISTORBDATA.....	6
3.14. LISTORBVOTDATA	7
3.15. LISTORBDATA2.....	7
3.16. LISTATTDATA	7
3.17. LISTATTVOTDATA	7
3.18. LISTEMEATTDATA.....	8
3.19. LISTATTDATA2	8
3.20. LISTNEWFRAMEORB	9
3.21. LISTNEWFRAMEATT.....	9
3.22. LISTRTNFRAMEATT	10
3.23. GETORBURL	10
3.24. PUTRESINRESP	10

3.25. CALCULATESURFACEFROMCUBE.....	11
3.26. CALCULATEDATAPOINTVALUEINCUBE	11
3.27. LISTCMEFRONTS.....	11
3.28. LISTCRMAPTYPES	12
3.29. LISTCRMAPS.....	12
3.30. LISTPROJMAPS	12
4. MODELS SERVICE	13
4.1. CALCULATEDATAPOINTVALUEYSY96	13
4.2. CALCULATEFIELDLINETS96	14
4.3. CALCULATEDATAPOINTVALUECAIN	14
4.4. CALCULATEFIELDLINECAIN	14
5. DATA TYPES USED IN SERVICES	15

GLOSSARY AND LIST OF TBC AND TBD ITEMS

AMDA	Automated Multi Dataset Analysis
CDPP	<i>Centre de Données de Physique des Plasmas</i>
SINP	Skobeltsyn Institute of Nuclear Physics
SOAP	Simple Object Access Protocol
URL	Uniform Resource Locator

List of TBC items:

List of TBD items:

1.OVERVIEW

1.1.APPLICABLE DOCUMENTS

DA1 Cf. les DA du Répertoire de la documentation 3DView/CDPP
J.-P. TONIUTTI, 22/06/2016, Issue 00, Rev. 06
CDPP-LI-32600-532-GFI

1.2.REFERENCE DOCUMENTS

DR1 Cf. les DA du Répertoire de la documentation 3DView/CDPP
J.-P. TONIUTTI, 22/06/2016, Issue 00, Rev. 06
CDPP-LI-32600-532-GFI

2.INTRODUCTION

This document describes 3DView CDPP web service interfaces.

3DView CDPP provides services for orbit/attitude data, coordinate system changes and magnetic field lines from models.

Web services use SOAP Document/Literal protocol.

3.3DVIEW FUNCTIONNAL WEB SERVICES

These services are used by 3DView client side but they are available for everyone.

Definition is available here: <http://3dview.cdpp.eu/CdppServices?wsdl>

3.1.LISTFRAMES

Java method definition: List<Frame> listFrames()

Returns the list of available frames for methods listOrbData, listOrbVOTData, ListAttData and ListAttVOTData.

3.2.LISTFRAMES2

Java method definition: List<Frame> listFrames2()

Returns the list of available frames for methods listNewFrameAtt, listNewFrameOrb, listOrbData2 and ListAttData2.

3.3.LISTBODIES

Java method definition: List<Body> listBodies(BodyType pType)

Returns all available bodies, natural or artificial, depending on pType parameter. If pType set to null, all bodies are returned.

pType is one of SPACECRAFT, PLANET, SATELLITE, COMET, ASTEROID.

3.4.LISTBODIESINREGION

Java method definition: List<Body> listBodiesInRegion(int pCenterNaifId, long pDist, List<Integer> pLstNaifIds, int pMinNaifId, int pMaxNaifId, XMLGregorianCalendar pDateStart, XMLGregorianCalendar pDateStop, List<TimeFile> pTimeFiles)

Returns all available bodies, natural or artificial, near from a center body at a defined distance between 2 dates.

pCenterNaifId: Center body id.

pDist: distance from center.

pLstNaifIds: Restricted research to a list of bodies

pMinNaifId: Min body id

pMaxNaifId: Max body id.

pDateStart: start date of search.

pDateStop: end date of search.

pTimeFiles: optional list of TimeFile containing body to load.

3.5.LISTBOIS

Java method definition: List<Body> listBois(int pBodyId)

List natural bodies of interest associated to a spacecraft.

For Rosetta, bodies of interest are Earth, Mars, 67P/CG, Steins and Lutetia.

3.6.LISTFILES

Java method definition: List<TimeFile> listFiles(int pBodyId, TimeFileType type, Date pStartTime, Date pStopTime)

List available files of a spacecraft, covering specified time range. It must be used before listOrbData and listAttData call.

TimeFileType is one of ORBIT, ATTITUDE, SA (solar array), HGA (High gain antenna) and EVENT.

3.7.GETCOVERAGE

Java method definition: TimeRange getCoverage(int pBodyId)

Get overall data coverage for a body.

3.8.GETSIZE

```
Tuple3D getSize(int pBodyId)
```

Get 3D radius size of a natural body in km.

3.9.LISTSTARSUBSETS

```
Java method definition: List<StarSubset> listStarSubsets()
```

List available star catalog. Use it before calling listStars.

3.10.LISTSTARS

```
Java method definition: List<Star> listStars(int pStarSubsetId)
```

Get stars of one subset got from listStarSubsets method. A star contains its Id, declination, right ascension and visual magnitude.

3.11.LISTINSTS

```
Java method definition: List<Instrument> listInsts(int pBodyId)
```

Returns instruments of a spacecraft.

3.12.LISTGS

```
Java method definition: List<GroundStation> listGS(int pBodyId)
```

Returns earth ground stations of a spacecraft.

3.13.LISTORBDATA

```
Java method definition: List<Tuple3D> listOrbData(TimeFile pFile, int pBodyId,int pFrameId, int pCenterId, XMLGregorianCalendar pStartTime, XMLGregorianCalendar pStopTime, int pStep)
```

Get orbit data of a body from its naif id in km.

For a spacecraft, a TimeFile of type ORBIT must be provided. For natural body, leave it null.

pStep must be provided in seconds.

3.14.LISTORBVOTDATA

Java method definition: String listOrbVOTData(TimeFile pFile, int pBodyId,int pFrameId, int pCenterId, XMLGregorianCalendar pStartTime, XMLGregorianCalendar pStopTime, int pStep)

Same as listOrbData but returns a URL pointing to a VOTable formatted file.

3.15.LISTORBDATA2

Java method definition: String listOrbData2(int pBodyId, String pFrame, List<XMLGregorianCalendar> pTimes, List<TimeFile> pTimeFiles)

Get Orbit data with new frames implementation.

pBodyId: Naif Id of body we want the orbit

pFrame: Coordinate system of orbit values

pTimes: time values for which we want values.

pTimeFiles: optional list of TimeFile containing body orbit to load.

Returns a URL pointing to a VOTable formatted file with orbit values in km.

3.16.LISTATTDATA

Java method definition: List<Matrix3D> listAttData(TimeFile pFile, int pBodyId, int pFrameId, XMLGregorianCalendar pStartTime, XMLGregorianCalendar pStopTime, int pStep)

Returns rotation attitude matrix of a frame relative to another.

For a spacecraft, a TimeFile of type ATTITUDE must be provided. For natural body, leave it null.

pStep must be provided in seconds.

3.17.LISTATTVOTDATA

Java method definition: String listAttVOTData(TimeFile pFile, int pBodyId, int pFrameId, XMLGregorianCalendar pStartTime, XMLGregorianCalendar pStopTime, int pStep)

Same as listOrbData but returns a URL pointing to a VOTable formatted file.

3.18.LISTEMEATTDATA

Java method definition: List<Matrix3D> listEmeAttData(int pFrameId, XMLGregorianCalendar pStartTime, XMLGregorianCalendar pStopTime, int pStep)

Returns list of rotation matrix from specified attitude to EMEJ2000 frame.

pStep must be provided in seconds.

3.19.LISTATTDATA2

Java method definition: String listAttData2(int pBodyId, String pFrame, List<XMLGregorianCalendar> pTimes, List<TimeFile> pTimeFiles)

Get Attitude data with new frames implementation.

pBodyId: Naif Id of body or frame we want the attitude

pFrame: Coordinate system of frame output values

pTimes: time values for which we want values.

pTimeFiles: optional list of TimeFile containing body/frame attitude to load.

Returns a URL pointing to a VOTable formatted file with rotation matrices.

3.20.LISTNEWFRAMEORB

Java method definition: String listNewFrameOrb(List< XMLGregorianCalendar > pDateTimeInput, List<Tuple3D> pDataInput, String pUrlInput, String pInFrame, String pInCenter, String pOutFrame, String pOutCenter, List<TimeFile> pTimeFiles)

Convert position values from a frame to another. It performs rotation and translation transforms.

Two calls are available: giving a URL pointing to a VOTable formatted file or a list of points(Tuple3D).

Values are in km.

pTimes: list of times corresponding to input values.

pValues: Values to transform in km. Must have same length than pTimes list.

pUrlInput: File containing times and values. If not null, pTimes and pValues are not used.

pInFrame: Coordinate system of frame input values

pInCenter: For future use

pOutFrame: Coordinate system of frame output values

pOutCenter: For future use

pTimeFiles: optional list of TimeFile containing body/frame attitude to load. Can be null.

Returns a URL pointing to a VOTable formatted file. If a frame unknown, an error is sent in returned String.

3.21.LISTNEWFRAMEATT

Java method definition: String listNewFrameAtt(List< XMLGregorianCalendar > pTimes, List<Tuple3D> pValues, String pUrlInput, String pInFrame, String pInCenter, String pOutFrame, String pOutCenter, List<TimeFile> pTimeFiles)

Convert vector values from a frame to another. It performs only rotation transforms.

Two calls are available: giving a URL pointing to a VOTable formatted file or a list of points (Tuple3D).

pTimes: list of times corresponding to input values.

pValues: Values to transform in km. Must have same length than pTimes list.

pUrlInput: URL of a VOTable containing times and values. If not null, pTimes and pValues are not used.

pInFrame: Coordinate system of frame input values

pInCenter: For future use

pOutFrame: Coordinate system of frame output values

pOutCenter: For future use

pTimeFiles: optional list of TimeFile containing body/frame attitude to load. Can be null.

Returns a URL pointing to a VOTable formatted file. If a frame unknown, an error message is sent in returned String.

3.22.LISTRNFRAMEATT

Java method definition: String listRtnFrameAtt(String pUrlInput, String plnFrame, int pOutFrameCenterId, String pOutFramePosUrl, int pOutFrameBodyId, List<TimeFile> pTimeFiles)

Get Attitude data for a RTN type frame given a center known body and a list of position or a body. It can be used for RTN or RTP frames.

R is positive from the center body to the given positions.

T is omega cross R, where omega is the center body spin axis.

N is R cross T, which completes the right-handed system.

pUrlInput: URL of a VOTable with input time/values to transform

plnFrame: Frame of input values

pOutFrameCenterId: Naif Id of output frame center body

pOutFramePosUrl: URL of a VOTable containing list of time/position of object/observatory

pOutFrameBodyId: Naif Id of object/observatory. If object positions are unknown natively (i.e. spacecraft), TimeFiles must be added with pTimeFiles List. If not null, pFramePosUrl is ignored.

pTimeFiles: optional list of TimeFile containing spice data to load.

Returns a URL pointing to a VOTable formatted file. If a frame unknown, an error message is sent in returned String.

Not implemented yet.

3.23.GETORBURL

Java method definition: String getOrbUrl(List< XMLGregorianCalendar > pDateTimelInput, List<Tuple3D> pDataInput)

Generates a VOTable file from a list of positions and returns a URL pointing to it.

3.24.PUTRESINRESP

Java method definition: boolean putResInRep(eu.cdpp._3dview.PutResInRep parameters)

Parameters:

String resourceID; // resource id of Cube already saved in server cache

String url; //file url to download

String provider; //resource provider

String frame; //coordinate system of cube

Put a cube resource into server repository for later processing with calculateSurfaceFromCube or calculateDataPointValueInCube.

3.25. CALCULATESURFACEFROMCUBE

Java method definition: String calculateSurfaceFromCube(eu.cdpp._3dview.CalculateSurfaceFromCube parameters)

Parameters:

String resourceID; // resource id of Cube already saved in server cache
List<String> params; // list of parameters to extract from cube
Tuple3D normal; // Surface normal vector
Tuple3D point; // point in surface to extract
float step; // grid step in km
OutputFormat outputFileType; // ZVOTable or VOTable

Generates a surface from a cube and returns an URL pointing to it.

3.26. CALCULATEDATAPOINTVALUEINCUBE

Java method definition: String calculateDataPointValueInCube(eu.cdpp._3dview.CalculateDataPointValueInCube parameters)

Parameters:

String resourceID; // resource id of Cube already saved in server cache
List<String> params; // list of parameters to extract from cube
String urlXYZ; // list of input points coordinates
OutputFormat outputFileType; // ZVOTable or VOTable

Generates a VOTable file containing the positions and the interpolated values at the given positions in the coordinates system of the simulation from a cube and a list of positions. It returns a URL pointing to the VOTable file.

3.27. LISTCMEFRONTS

Java method definition: String listCMEFronts(String pObserverName, XMLGregorianCalendar pStartTime, XMLGregorianCalendar pStopTime)

Get CME Fronts data from Helcats catalogue between 2 dates.

pObserverName: Name of observer body of the CME fronts

pStartTime: start time for which we want values.

pEndTime: end time for which we want values.

Returns a URL pointing to an ASCII file containing the list of data.

3.28.LISTCRMAPTYPES

Java method definition: List<CarringtonMapType> listCRMapTypes()

Get list of Carrington map types available in Propagation tool Carrington catalogue.

Returns a list of Carrington map type.

3.29.LISTCRMAPS

Java method definition: List<CarringtonMap> listCRMMaps(XMLGregorianCalendar pStartTime, XMLGregorianCalendar pStopTime, String pPath)

Get list of Carrington maps between start time and stop time for a specific.

pStartTime: start time for which we want values.

pEndTime: end time for which we want values.

pPath: path within the Propagation tool Carrington catalogue.

Returns a list of Carrington maps containing an URL pointing to a png file and a start time.

3.30.LISTPROJMAPS

Java method definition: List<ProjectionMap> listProjMaps(int pNaifId)

Get list of Projection maps of a body.

pNaifId: Naif ID on which the maps has to be projected.

Returns a list of Projection maps containing an URL pointing to a png file.

4.MODELS SERVICE

3DView provides a service for tracing the magnetic field lines. This service will use models of magnetic field (to be implemented in 3DView) of Tsyganenko, Cain and the stable models of SINP.

The Tsyganenko and IGRF models will be used for the external and internal field computation. The external magnetic field is calculated from the model of Tsyganenko T96.

This model has an explicitly defined realistic magnetopause, large-scale Region 1 and 2 Birkeland current systems, and IMF penetration across the boundary. The model takes as input the solar wind ram pressure, the DST index, the transverse components of the interplanetary magnetic field (By, Bz), the dipole tilt of the Earth and the position (x, y, z) in GSM coordinates. The solar wind data and DST will be provided by an AMDA web service. The Earth's magnetic field is calculated in GSM coordinates.

The Cain model will calculate the magnetic field of Mars. Cain proposes a model of crustal magnetic field obtained from a potential function using spherical harmonics up to degree $n = 90$. The model provides the Schmidt coefficients for the calculation of the magnetic field in a planetographic reference frame (longitude, latitude, altitude).

The available simulationRuns to access Web services can be found in tree http://3dview.cdpp.eu/other/3dview_tree.xml

Definition of service is available here: http://3dview.cdpp.eu/Methods_3DVIEW?wsdl

4.1.CALCULATEDATAPOINTVALUETSY96

Java method definition: String calculateDataPointValueTsy96 (CalculateDataPointValueTsy96 parameters)

Parameters:

String resourceID; // resource id of simulationRun

String urlXYZ; // list of input points coordinates

ExtraParamsCalculateEarth extraParamsEarth; Earth weather inputs(BIMF, SW pressure, ...)

OutputFormat outputFileType; //ZVOTable or VOTable

Returns magnetic field for given 3D points provided by user. User can also specify physical parameters like Solar wind pressure, speed and IMF.

Physical inputs:

Name	Description	Unit	Min	Max
SWPressure	Solar wind pressure	nPa	0.5	10
DST	Activity index	nT	-100	20
IMF	Magnetic field IMF in GSM. Set only y and z components.	nT	(0,-10,-10)	(0,10,10)

4.2.CALCULATEFIELDLINETS96

Java method definition: String calculateFieldLineTsy96 (calculateFieldLineTsy96 parameters)

Parameters:

String resourceID; // resource id of simulationRun

String urlXYZ; // list of input points coordinates

ExtraParamsCalculateFieldLine extraParamsFieldLine; //Field line parameters (Number of points, line length)

ExtraParamsCalculateEarth extraParamsEarth; //Earth weather inputs(BIMF, SW pressure, ...)

OutputFormat outputFileType; //ZVOTable or VOTable

Returns magnetic field lines for given set of initial 3D points provided by user. User can also specify physical parameters like Solar wind pressure, speed and IMF.

Physical inputs:

Name	Description	Unit	Min	Max
SWPressure	Solar wind pressure	nPa	0.5	10
DST	Activity index	nT	-100	20
IMF	Magnetic field IMF in GSM. Set only y and z components.	nT	(0,-10,-10)	(0,10,10)

4.3.CALCULATEDATAPOINTVALUECAIN

Java method definition: String calculateDataPointValueCain (CalculateDataPointValueCain parameters)

Parameters:

String resourceID; // resource id of simulationRun

String urlXYZ; // list of input points coordinates

OutputFormat outputFileType; //ZVOTable or VOTable

Returns magnetic field for given 3D points provided by user.

4.4.CALCULATEFIELDLINECAIN

Java method definition: String calculateFieldLineCain (CalculateFieldLineCain parameters)

Parameters:

String resourceID; // resource id of simulationRun

String urlXYZ; // list of input points coordinates

ExtraParamsCalculateFieldLine extraParamsFieldLine; //Field line parameters (Number of points, line length)

OutputFormat outputFileType; //ZVOTable or VOTable

Returns magnetic field lines for given set of initial 3D points provided by user.

5. DATA TYPES USED IN SERVICES

Data types exchanged in services are defined in an xsd files.

Methods parameters are contained in cdpp3dview_params.xsd:

```
<xsd:schema version="1.0" targetNamespace="http://cdpp3dview/params"
xmlns:tns="http://cdpp3dview/params"
xmlns:typ="http://cdpp3dview/types"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:jaxws="http://java.sun.com/xml/ns/jaxws"
xmlns:jaxb="http://java.sun.com/xml/ns/jaxb" jaxb:version="2.0">

  <xsd:import namespace="http://cdpp3dview/types" schemaLocation="cdpp3dview_types.xsd"/>
  <xsd:element name="listFrames">
    <xsd:complexType><xsd:sequence/></xsd:complexType>
  </xsd:element>
  <xsd:element name="listFramesResponse">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element minOccurs="0" maxOccurs="unbounded" name="return" type="typ:Frame"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="listFrames2">
    <xsd:complexType><xsd:sequence/></xsd:complexType>
  </xsd:element>
  <xsd:element name="listFrames2Response">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element minOccurs="0" maxOccurs="unbounded" name="return" type="typ:Frame"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="listBodies">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="pType" type="typ:BodyType"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="listBodiesResponse">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element minOccurs="0" maxOccurs="unbounded" name="return" type="typ:Body"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
```

```

<xsd:element name="listBois">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="pBodyId" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="listBoisResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element minOccurs="0" maxOccurs="unbounded" name="return" type="typ:Body"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="listFiles">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="pBodyId" type="xsd:int" nillable="false"/>
      <xsd:element name="type" type="typ:TimeFileType" nillable="true"/>
      <xsd:element name="pStartTime" type="xsd:dateTime" nillable="false"/>
      <xsd:element name="pStopTime" type="xsd:dateTime" nillable="false"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="listFilesResponse" type="typ:TimeFileArray"/>

<xsd:element name="getCoverage">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="pBodyId" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:complexType name="getCoverageResponse">
  <xsd:sequence>
    <xsd:element name="return" type="typ:TimeRange"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="getCoverageResponse" type="tns:getCoverageResponse"/>
<xsd:element name="getSize">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="pBodyId" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:complexType name="getSizeResponse">
  <xsd:sequence>
    <xsd:element name="return" type="typ:Tuple3d"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="getSizeResponse" type="tns:getSizeResponse"/>

```

```

<xsd:element name="listStarSubsets">
  <xsd:complexType><xsd:sequence/></xsd:complexType>
</xsd:element>
<xsd:complexType name="listStarSubsetsResponse">
  <xsd:sequence>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="starSubset" type="typ:StarSubset"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="listStarSubsetsResponse" type="tns:listStarSubsetsResponse"/>

<xsd:element name="listStars">
  <xsd:complexType><xsd:sequence>
    <xsd:element name="pStarSubsetId" type="xsd:int"/>
  </xsd:sequence></xsd:complexType>
</xsd:element>
<xsd:complexType name="listStarsResponse">
  <xsd:sequence>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="star" type="typ:Star"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="listStarsResponse" type="tns:listStarsResponse"/>

<xsd:element name="listInsts">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="pBodyId" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:complexType name="listInstsResponse">
  <xsd:sequence>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="star" type="typ:Instrument"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="listInstsResponse" type="tns:listInstsResponse"/>

<xsd:element name="listGS">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="pBodyId" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:complexType name="listGSResponse">
  <xsd:sequence>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="star" type="typ:GroundStation"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="listGSResponse" type="tns:listGSResponse"/>

<xsd:element name="listOrbData">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="pFile" type="typ:TimeFile"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

```

```

    <xsd:element name="pBodyId" type="xsd:int"/>
    <xsd:element name="pFrameId" type="xsd:int"/>
    <xsd:element name="pCenterId" type="xsd:int"/>
    <xsd:element name="pStartTime" type="xsd:dateTime"/>
    <xsd:element name="pStopTime" type="xsd:dateTime"/>
    <xsd:element name="pStep" type="xsd:int"/>
  </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="listOrbDataResponse" type="typ:Tuple3dArray"/>

<xsd:element name="listOrbVOTData">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="pFile" type="typ:TimeFile"/>
      <xsd:element name="pBodyId" type="xsd:int"/>
      <xsd:element name="pFrameId" type="xsd:int"/>
      <xsd:element name="pCenterId" type="xsd:int"/>
      <xsd:element name="pStartTime" type="xsd:dateTime"/>
      <xsd:element name="pStopTime" type="xsd:dateTime"/>
      <xsd:element name="pStep" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="listOrbVOTDataResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="return" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="listOrbData2">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element minOccurs="0" maxOccurs="unbounded" name="pBodyId" type="xsd:int"/>
      <xsd:element name="pFrame" type="typ:Frame"/>
      <xsd:element name="pCenterId" type="xsd:int"/>
      <xsd:element minOccurs="0" maxOccurs="unbounded" name="pTimes" type="xsd:dateTime"/>
      <xsd:element minOccurs="0" maxOccurs="unbounded" name="pTimeFiles" type="typ:TimeFile"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="listOrbData2Response">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element minOccurs="0" maxOccurs="unbounded" name="return" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

```



```

<xsd:element name="listAttData">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="pFile" type="typ:TimeFile"/>
      <xsd:element name="pBodyId" type="xsd:int"/>
      <xsd:element name="pFrameId" type="xsd:int"/>
      <xsd:element name="pStartTime" type="xsd:dateTime"/>
      <xsd:element name="pStopTime" type="xsd:dateTime"/>
      <xsd:element name="pStep" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="listAttDataResponse" type="typ:Matrix3dArray"/>

<xsd:element name="listAttVOTData">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="pFile" type="typ:TimeFile"/>
      <xsd:element name="pBodyId" type="xsd:int"/>
      <xsd:element name="pFrameId" type="xsd:int"/>
      <xsd:element name="pStartTime" type="xsd:dateTime"/>
      <xsd:element name="pStopTime" type="xsd:dateTime"/>
      <xsd:element name="pStep" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="listAttVOTDataResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="return" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="listEmeAttData">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="pFrameId" type="xsd:int"/>
      <xsd:element name="pStartTime" type="xsd:dateTime"/>
      <xsd:element name="pStopTime" type="xsd:dateTime"/>
      <xsd:element name="pStep" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="listEmeAttDataResponse" type="typ:Matrix3dArray"/>

<xsd:element name="listAttData2">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element minOccurs="0" maxOccurs="unbounded" name="pBodyId" type="xsd:int"/>
      <xsd:element name="pFrame" type="typ:Frame"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

```

```

    <xsd:element minOccurs="0" maxOccurs="unbounded" name="pTimes" type="xsd:dateTime"/>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="pTimeFiles" type="typ:TimeFile"/>
  </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="listAttData2Response">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element minOccurs="0" maxOccurs="unbounded" name="return" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="listNewFrameOrb">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element minOccurs="0" maxOccurs="unbounded" name="pDateTimelInput" type="xsd:dateTime"/>
      <xsd:element minOccurs="0" maxOccurs="unbounded" name="pDataInput" type="typ:Tuple3d"/>
      <xsd:element name="pUrlInput" nillable="true" type="xsd:string"/>
      <xsd:element name="pInFrame" type="xsd:string"/>
      <xsd:element name="pInCenter" type="xsd:string"/>
      <xsd:element name="pOutFrame" type="xsd:string"/>
      <xsd:element name="pOutCenter" type="xsd:string"/>
      <xsd:element minOccurs="0" maxOccurs="unbounded" name="pTimeFiles" type="typ:TimeFile"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="listNewFrameOrbResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="return" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="listNewFrameAtt">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element minOccurs="0" maxOccurs="unbounded" name="pDateTimelInput" type="xsd:dateTime"/>
      <xsd:element minOccurs="0" maxOccurs="unbounded" name="pDataInput" type="typ:Tuple3d"/>
      <xsd:element name="pUrlInput" nillable="true" type="xsd:string"/>
      <xsd:element name="pInFrame" type="xsd:string"/>
      <xsd:element name="pInCenter" type="xsd:string"/>
      <xsd:element name="pOutFrame" type="xsd:string"/>
      <xsd:element name="pOutCenter" type="xsd:string"/>
      <xsd:element minOccurs="0" maxOccurs="unbounded" name="pTimeFiles" type="typ:TimeFile"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="listNewFrameAttResponse">
  <xsd:complexType>
    <xsd:sequence>

```

```

    <xsd:element name="return" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
</xsd:element>

<xsd:element name="listRtnFrameAtt">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="pUrlInput" type="xsd:string"/>
      <xsd:element name="pInFrame" type="xsd:string"/>
      <xsd:element name="pOutFrameCenterId" type="xsd:int"/>
      <xsd:element name="pOutFramePosUrl" type="xsd:string"/>
      <xsd:element name="pOutFrameBodyId" type="xsd:int"/>
      <xsd:element minOccurs="0" maxOccurs="unbounded" name="pTimeFiles" type="typ:TimeFile"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="listRtnFrameAttResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="return" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="getOrbUrl">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element minOccurs="0" maxOccurs="unbounded" name="pDateTimeInput" type="xsd:dateTime"/>
      <xsd:element minOccurs="0" maxOccurs="unbounded" name="pDataInput" type="typ:Tuple3d"/>
      <xsd:element name="pUnit" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="getOrbUrlResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="return" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="listProjMaps">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="pNaifId" type="xsd:int" nillable="false"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="listProjMapsResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element minOccurs="0" maxOccurs="unbounded" name="projectionMap" type="typ:ProjectionMap"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

```

```
</xsd:sequence>
</xsd:complexType>
</xsd:element>

<xsd:element name="listCRMapTypes">
  <xsd:complexType><xsd:sequence/></xsd:complexType>
</xsd:element>
<xsd:element name="listCRMapTypesResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element minOccurs="0" maxOccurs="unbounded" name="carringtonMapType"
type="typ:CarringtonMapType"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="listCRMaps">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="pStartTime" type="xsd:dateTime"/>
      <xsd:element name="pStopTime" type="xsd:dateTime"/>
      <xsd:element name="pPath" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="listCRMapsResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element minOccurs="0" maxOccurs="unbounded" name="carringtonMap" type="typ:CarringtonMap"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="listBodiesInRegion">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="pCenterNaifId" nillable="false" type="xsd:int"/>
      <xsd:element name="pDist" nillable="false" type="xsd:long"/>
      <xsd:element minOccurs="0" maxOccurs="unbounded" name="pLstNaifIds" type="xsd:int"/>
      <xsd:element name="pMinNaifId" nillable="false" type="xsd:int"/>
      <xsd:element name="pMaxNaifId" nillable="false" type="xsd:int"/>
      <xsd:element name="pDateStart" nillable="false" type="xsd:dateTime"/>
      <xsd:element name="pDateStop" nillable="false" type="xsd:dateTime"/>
      <xsd:element name="pTimeFiles" minOccurs="0" maxOccurs="unbounded" nillable="false" type="typ:TimeFile"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="listBodiesInRegionResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element minOccurs="0" maxOccurs="unbounded" name="return" type="typ:Body"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

```

<xsd:element name="listCMEFronts">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="pObserverName" type="xsd:string"/>
      <xsd:element name="pStartTime" type="xsd:dateTime"/>
      <xsd:element name="pStopTime" type="xsd:dateTime"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="listCMEFrontsResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="return" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

```

Parameters types are defined in cdpp3dview_types.xsd:

```

<xsd:schema version="1.0" targetNamespace="http://cdpp3dview/types"
  xmlns:tns="http://cdpp3dview/types"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:jaxws="http://java.sun.com/xml/ns/jaxws"
  xmlns:jaxb="http://java.sun.com/xml/ns/jaxb" jaxb:version="2.0">

  <xsd:complexType name="TimeRange">
    <xsd:sequence>
      <xsd:element name="startTime" type="xsd:dateTime"/>
      <xsd:element name="stopTime" type="xsd:dateTime"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="Frame">
    <xsd:sequence>
      <xsd:element name="id" type="xsd:int"/>
      <xsd:element name="name" type="xsd:string"/>
      <xsd:element name="desc" type="xsd:string" nillable="true"/>
      <xsd:element minOccurs="0" maxOccurs="unbounded" name="center" type="tns:Body"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="StarSubset">
    <xsd:sequence>
      <xsd:element name="id" type="xsd:int"/>
      <xsd:element name="name" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="Star">
    <xsd:sequence>
      <xsd:element name="id" type="xsd:int"/>
      <xsd:element name="dec" type="xsd:float"/>
      <xsd:element name="ra" type="xsd:float"/>
      <xsd:element name="vm" type="xsd:float"/>
    </xsd:sequence>
  </xsd:complexType>

```

```

</xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="BodyType" >
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="SPACECRAFT"/>
    <xsd:enumeration value="PLANET"/>
    <xsd:enumeration value="SATELLITE"/>
    <xsd:enumeration value="COMET"/>
    <xsd:enumeration value="ASTEROID"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="TimeFileType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="ORBIT"/>
    <xsd:enumeration value="ATTITUDE"/>
    <xsd:enumeration value="SA"/>
    <xsd:enumeration value="HGA"/>
    <xsd:enumeration value="EVENT"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="InstrumentRep">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="CONE"/>
    <xsd:enumeration value="LINE"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="Body">
  <xsd:sequence>
    <xsd:element name="naifId" type="xsd:int"/>
    <xsd:element name="modelId" type="xsd:int"/>
    <xsd:element name="name" type="xsd:string"/>
    <xsd:element name="coverage" type="tns:TimeRange"/>
    <xsd:element name="type" type="tns:BodyType"/>
    <xsd:element name="color" type="xsd:string"/>
    <xsd:element name="size" type="tns:Tuple3d"/>
    <xsd:element name="prefFrame" type="xsd:int"/>
    <xsd:element name="prefCenter" type="xsd:int"/>
    <xsd:element name="prefStarSubset" type="xsd:int"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="TimeFile">
  <xsd:sequence>
    <xsd:element name="naifId" type="xsd:int"/>
    <xsd:element name="name" type="xsd:string"/>
    <xsd:element name="startTime" type="xsd:dateTime"/>
    <xsd:element name="stopTime" type="xsd:dateTime"/>
    <xsd:element name="type" type="tns:TimeFileType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Instrument">
  <xsd:sequence>
    <xsd:element name="naifId" type="xsd:int"/>
    <xsd:element name="name" type="xsd:string"/>

```

```

<xsd:element name="repr" type="tns:InstrumentRep"/>
<xsd:element name="color" type="xsd:string"/>
<xsd:element name="vBSight" type="tns:Tuple3d"/>
<xsd:element name="vUp" type="tns:Tuple3d"/>
<xsd:element name="xFovDeg" type="xsd:float"/>
<xsd:element name="yFovDeg" type="xsd:float"/>
<xsd:element name="fronClipDist" type="xsd:float"/>
<xsd:element name="backClipDist" type="xsd:float"/>
<xsd:element name="isAdaptative" type="xsd:boolean"/>
<xsd:element name="targetOrLength" type="xsd:float"/>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="GroundStation">
  <xsd:sequence>
    <xsd:element name="naifId" type="xsd:int"/>
    <xsd:element name="name" type="xsd:string"/>
    <xsd:element name="ra" type="xsd:float"/>
    <xsd:element name="dec" type="xsd:float"/>
    <xsd:element name="color" type="xsd:string"/>
    <xsd:element name="angle" type="xsd:float"/>
    <xsd:element name="heightKm" type="xsd:float"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ProjectionMap">
  <xsd:sequence>
    <xsd:element name="naifId" type="xsd:int"/>
    <xsd:element name="name" type="xsd:string"/>
    <xsd:element name="desc" type="xsd:string"/>
    <xsd:element name="projType" type="xsd:string">
      <xsd:annotation><xsd:documentation>
        Projection type, mercator, cassini, ...
      </xsd:documentation></xsd:annotation>
    </xsd:element>
    <xsd:element name="altitude" type="xsd:float">
      <xsd:annotation><xsd:documentation>
        Altitude of map in km from center.
      </xsd:documentation></xsd:annotation>
    </xsd:element>
    <xsd:element name="url" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="CarringtonMapType">
  <xsd:sequence>
    <xsd:element name="observatory" type="xsd:string"/>
    <xsd:element name="instrument" type="xsd:string"/>
    <xsd:element name="measurement" type="xsd:string"/>
    <xsd:element name="path" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="CarringtonMap">
  <xsd:sequence>

```

```
<xsd:element name="startTime" type="xsd:dateTime"/>
<xsd:element name="url" type="xsd:string"/>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Tuple3d">
  <xsd:sequence>
    <xsd:element name="x" type="xsd:double" nillable="false"/>
    <xsd:element name="y" type="xsd:double" nillable="false"/>
    <xsd:element name="z" type="xsd:double" nillable="false"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Tuple6d">
  <xsd:sequence>
    <xsd:element name="x" type="xsd:double" nillable="false"/>
    <xsd:element name="y" type="xsd:double" nillable="false"/>
    <xsd:element name="z" type="xsd:double" nillable="false"/>
    <xsd:element name="vx" type="xsd:double" nillable="false"/>
    <xsd:element name="vy" type="xsd:double" nillable="false"/>
    <xsd:element name="vz" type="xsd:double" nillable="false"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Matrix3d">
  <xsd:sequence>
    <xsd:element minOccurs="9" maxOccurs="9" name="values" type="xsd:double"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="TimeFileArray">
  <xsd:sequence>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="timeFile" type="tns:TimeFile"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="DateTimeArray">
  <xsd:sequence>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="dateTime" type="xsd:dateTime"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Tuple3dArray">
  <xsd:sequence>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="tuple3d" type="tns:Tuple3d"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Matrix3dArray">
  <xsd:sequence>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="matrix3d" type="tns:Matrix3d"/>
  </xsd:sequence>
</xsd:complexType>
```