IMPEX

IMPEx-IF-1-2-GFI
Issue : 00 Date : 28/03/2013
Revision : 04 Date : 24/01/2014
Ref. : IRAP/IMPEX/PRD/DOC/IF MT : X Distribution Code : E

INTERFACES SPECIFICATIONS
IMPEX

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

1, passage de l'Europe 31400 Toulouse - Tél. 05 61 75 00 40

INDEX SHEET

CONFIDENTIALITY: NC	KEYWORDS: 3DView, Web Services, IMPEx, Interface, Specification

TITLE:

Interfaces Specifications

IMPEx

AUTHOR(S) :

BEIGBEDER Laurent **GFI INFORMATIQUE**

SUMMARY: This document describes the software interfaces

RELATED DOCUMENTS: Stand alone document.	LOCALIZATION: IRAP/IMPEX/PRD/DOC

VOLUME: 1	TOTAL NUMBER OF PAGES: 23 INCLUDING PRELIMINARY PAGES: 6 NUMBER OF SUPPL. PAGES: 0	COMPOSITE DOCUMENT : N	LANGUAGE : EN

CONFIGURATION MANAGEMENT : NG	CM RESP. :

REASONS FOR EVOLUTION : Update interfaces parameters

CONTRACT: DC3 - Acte d'engagement Marché 12.14.049

HOST SYSTEM:

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

INTERNAL DISTRIBUTION

Name	Entity	Internal Postal Box	Observations
GENOT Vincent	IRAP	CNRS	

EXTERNAL DISTRIBUTION

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

CHANGES

Issue	Rev.	Date	Reference, Author(s), Reasons for evolution
00	04	24/01/2014	IRAP/IMPEX/PRD/DOC/IF BEIGBEDER Laurent GFI INFORMATIQUE Update interfaces parameters
00	03	10/12/2013	IRAP/IMPEX/PRD/DOC/IF BEIGBEDER Laurent GFI INFORMATIQUE Update interfaces methods and Excel file with frames added
00	02	02/05/2013	IRAP/IMPEX/PRD/DOC/IF BEIGBEDER Laurent GFI INFORMATIQUE Taking comments into account IRAP / CNES
00	01	22/04/2013	IRAP/IMPEX/PRD/DOC/IF BEIGBEDER Laurent GFI INFORMATIQUE Taking comments into account IRAP / CNES (see reading sheet IRAP/IMPEX/RCL/REL/002)
00	00	28/03/2013	IRAP/IMPEX/PRD/DOC/IF BEIGBEDER Laurent GFI INFORMATIQUE Creation of the document

TABLE OF CONTENTS

GLOSSARY AND LIST OF TBC AND TBD ITEMS

AMDA	Automated Multi Dataset Analysis
CCTP	*Cahier des Clauses Techniques Particulières*
CDPP	*Centre de Données de Physique des Plasmas*
CNES	*Centre National d'Etudes Spatiales*
IMPEx	Integrated Medium for Planetary Exploration
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.REFERENCE DOCUMENTS

DR1 Document de conception globale du logiciel 3DView avec Web
Laurent BEIGBEDER, 02/05/2013, Issue. 00, Rev. 02
IMPEx-CG-1-1-GFI

DR2 Répertoire de la documentation IMPEx
Laurent BEIGBEDER, 12/12/2013, Édit. 00, Rév. 04
IMPEx-REP-1-9-GFI

DR3 List of IMPEx 3DView frames
Laurent BEIGBEDER, 10/12/2013, Édit. 00, Rév. 00
frames_IMPEx_3dview-GFI

1.2.APPLICABLE DOCUMENTS

AD1 SPECIFICATIONS DE BESOIN DE VISUALISATION DE DONNEES
Bruno BESSON, 30/11/2009, Issue 01, Rev. 01
CDPP-ST-32100-350-CNES

AD2 Evolutions multi-mission de 3DView
Henri Marquier, 15/03/2006, Issue 01, Rev. 00
3DVw-ST-0-1-CN

DA3 SPECIFICATIONS DE BESOIN EVOLUTIONS 3DVIEW
Bruno BESSON, 31/01/2011, Issue. 01, Rev. 00
CDPP-ST-32600-376-CNES

DA4 CDC Mise à jour de 3DView
Vincent GENOT, 06/06/2012, Édit. 01, Rév. 02
CCTP 3DView CDPP pour IMPEx n°12213 du 18/06/2012

DA5 Livret technique - Mise à jour de 3DView marché n° 12.14.049
GFI Informatique, 07/12/2012, Édit. 03, Rév. 00
GFI-12-n°32845

2.INTRODUCTION

This document describes 3DView Impex web service interfaces.

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

Web services use SOAP Document/Literal protocol.

Definition is available here: *http://3dview.cesr.fr/CdppServices?wsdl*

3.3DVIEW FUNCTIONNAL WEB SERVICES

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

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.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.

3.5.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.

3.6.GETCOVERAGE

```
Java method definition: TimeRange getCoverage(int pBodyId)
```

Get overall data coverage for a body.

3.7.GETSIZE

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

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

3.8.LISTSTARSUBSETS

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

List available star catalog. Use it before calling listStars.

3.9.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.

3.10.LISTINSTS

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

Returns instruments of a spacecraft.

3.11.LISTGS

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

Returns earth ground stations of a spacecraft.

3.12.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.

3.13.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.

3.14.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.

Not implemented yet.

3.15.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.

3.16.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.

3.17.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.

3.18.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.

Not implemented yet.

3.19.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.

Not implemented yet.

3.20.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.

3.21.LISTRTNFRAMEATT

```
Java method definition: String listRtnFrameAtt(String pUrlInput, String pInFrame, 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
pInFrame: 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.

3.22.GETORBURL

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

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

3.23.GETSURFACE

```
Java method definition: not available yet
```

Generates an iso-surface “obj” file from a cube and returns a URL pointing to it. The cube is a reference to a file in common space.

3.24.GETDATAPOINTVALUE

```
Java method definition: not available yet
```

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.

The cube is a reference to a file in common space.

4.MODELS SERVICE

```
Java method definition: not available yet
```

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 SINP provides an analytical model of the Earth's magnetic field. The stable version of this model will be implemented 3DView.

5.DATA TYPES USED IN SERVICES

Data types exchanged in services are defined in an xsd file :

```
    <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="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: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="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="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"/>
```

Enfin, la définition des types est dans le fichier cdpp3dview_types.xsd :

```
    <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 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: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="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="dir" 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="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="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>
```