The Consultative Committee for Space Data Systems

Proposed Recommendation for Space Data System Standards

POINTING REQUEST MESSAGE

PROPOSED STANDARD

CCSDS 509.0-W-2.6

WHITE BOOK

June 2015

AUTHORITY

Issue:	White Book, Issue 2.6
Date:	June 2015
Location:	Not Applicable

(WHEN THIS PROPOSED STANDARD IS FINALIZED, IT WILL CONTAIN THE FOLLOWING STATEMENT OF AUTHORITY:)

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and represents the consensus technical agreement of the participating CCSDS Member Agencies. The procedure for review and authorization of CCSDS documents is detailed in the *Procedures Manual for the Consultative Committee for Space Data Systems*, and the record of Agency participation in the authorization of this document can be obtained from the CCSDS Secretariat at the address below.

This document is published and maintained by:

CCSDS Secretariat
Space Communications and Navigation Office, 7L70
Space Operations Mission Directorate
NASA Headquarters
Washington, DC 20546-0001, USA

FOREWORD

This document is a Proposed Standard that has been prepared by the Consultative Committee for Space Data Systems (CCSDS). The message described in this Proposed Standard establishes a common framework and provides a common format for the interchange of data describing the request for pointing of a spacecraft. The standard was developed for specific use in applications that are cross-supported between Agencies of the CCSDS, but it is applicable to the activities of other space operators as well. It allows implementing organizations within each Agency to proceed coherently with the development of compatible derived standards for the flight and ground systems that are within their cognizance. Derived Agency standards may implement only a subset of the optional features allowed by the Proposed Standard and may incorporate features not addressed by this Proposed Standard.

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Proposed Standard is therefore subject to CCSDS document management and change control procedures, which are defined in the *Organization and Process for the Consultative Committee for Space Data Systems*. Current versions of CCSDS documents are maintained at the CCSDS Web site:

http://www.ccsds.org/

Questions relating to the contents or status of this document should be addressed to the CCSDS Secretariat at the address indicated on page i.

At time of publication, the active Member and Observer Agencies of the CCSDS were:

Member Agencies

- Agenzia Spaziale Italiana (ASI)/Italy.
- British National Space Centre (BNSC)/United Kingdom.
- Canadian Space Agency (CSA)/Canada.
- Centre National d'Etudes Spatiales (CNES)/France.
- China National Space Administration (CNSA)/People's Republic of China.
- Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Federal Space Agency (FSA)/Russian Federation.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- Japan Aerospace Exploration Agency (JAXA)/Japan.
- National Aeronautics and Space Administration (NASA)/USA.

Observer Agencies

- Austrian Space Agency (ASA)/Austria.
- Belgian Federal Science Policy Office (BFSPO)/Belgium.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- Centro Tecnico Aeroespacial (CTA)/Brazil.
- Chinese Academy of Sciences (CAS)/China.
- Chinese Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- Danish National Space Center (DNSC)/Denmark.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Hellenic National Space Committee (HNSC)/Greece.
- Indian Space Research Organization (ISRO)/India.

- Institute of Space Research (IKI)/Russian Federation.
- KFKI Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- Korea Aerospace Research Institute (KARI)/Korea.
- MIKOMTEK: CSIR (CSIR)/Republic of South Africa.
- Ministry of Communications (MOC)/Israel.
- National Institute of Information and Communications Technology (NICT)/Japan.
- National Oceanic and Atmospheric Administration (NOAA)/USA.
- National Space Organization (NSPO)/Chinese Taipei.
- Naval Center for Space Technology (NCST)/USA.
- Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.
- United States Geological Survey (USGS)/USA.

PREFACE

This document is a proposed CCSDS Proposed Standard. Its 'White Book' status indicates that the CCSDS believes the document is NOT technically mature, therefore it is only suitable for internal review by the CCSDS working group chartered to develop the recommendation. As such, its technical contents are not stable, and several iterations of it may occur in response to working group discussions.

Implementers are cautioned **not** to fabricate any final equipment in accordance with this document's technical content.

DOCUMENT CONTROL

Document	Title and Issue	Date	Status
CCSDS 509.0-W-1.0	Pointing Request Message, White Book 1.0	April 2011	Initial draft
CCSDS 509.0-W-2.0	Pointing Request Message, White Book 2.0	October 2011	Review for CCSDS after Fall Meeting 2011
CCSDS 509.0-W-2.1	Pointing Request Message, White Book 2.1	August 2012	Review for CCSDS after Spring Meeting 2012
CCSDS 509.0-W-2.2	Pointing Request Message, White Book 2.2	March 2013	Review for CCSDS after Fall Meeting 2012
CCSDS 509.0-W-2.3	Pointing Request Message, White Book 2.3	October 2013	Review for CCSDS after Spring Meeting 2013
CCSDS 509.0-W-2.4	Pointing Request Message, White Book 2.4	March 2013	Review for CCSDS after Fall Meeting 2013
CCSDS 509.0-W-2.5	Pointing Request Message, White Book 2.5	February 2015	Review for CCSDS after Srping Meeting 2014. Version for Spring Meeting 2015
CCSDS 509.0-W-2.5	Pointing Request Message, White Book 2.6	June 2015	Review for CCSDS after Srping Meeting 2015. Consolidation version with alignment with NDM/XML, definition of celestial bodies referencing.

CONTENTS

1 INTRODUCTION

1.1 PURPOSE

The Pointing Request Message (PRM) allows space agencies and operators to exchange information in a standardized format about a requested pointing of a spacecraft. These can be requested (sequences of) changes of the attitude of the spacecraft or of an articulated spacecraft component.

1.2 SCOPE

This Proposed Standard is applicable only to the message format and content, but not to its transmission. The method of transmitting the message between exchange partners could be based on a CCSDS data transfer protocol, a file based transfer protocol such as SFTP, stream-oriented media, or another secure transmission mechanism. In general, the transmission mechanism and the technical data content of a PRM are independent. It is recommended that the transmission method be documented in an Interface Control Document (ICD) between the exchange partners.

1.3 APPLICABILITY

The PRM facilitates interoperability between space agencies; e.g., where Agency/Operator A operates a spacecraft which provides a relay for a rover operated by Agency/Operator B or where an instrument owned and operated by Agency/Operator A is embarked on a spacecraft operated by Agency/Operator B. It can be used internally within a single agency or organization as well.

1.4 RATIONALE

It is necessary to formulate and to transmit pointing requests, but prior to this recommendation there was no formal standard for this purpose. Rather, pointing requests were formulated in natural language. Requests in natural language are imprecise, inefficient, and error prone. The purpose of the PRM is to formalize the way in which pointing requests are formulated and to facilitate their transmission and processing by automated means.

1.5 DOCUMENT STRUCTURE

Section 1 (this section) provides introductory matter.

Section 2 provides a brief technical overview of pointing requests.

Section 3 discusses the structure and content of the Pointing Request Message.

Section 3.1 provides a general introduction to the PRM structure.

Section 3.2 provides an overview of the PRM structure.

Section 3.3 specifies the XML elements available for constructing PRMs.

Section 3.4 specifies a definition and referencing mechanism which is fundamental to the PRM. It allows for covering the existing large spectrum of pointing scenarios in a compact and flexible manner by a single message. The need for this mechanism is the main reason why the PRM exists in XML notation only.

Section 4 specifies a normative set of templates for common, generic pointing scenarios. These templates can be referenced by mission specific ICDs where applicable.

Section 5 specifies rules for the construction of PRMs that are not covered by the generic templates provided in section 4.

ANNEX A provides the list of time systems and reference frames used.

ANNEX B lists a number of items to be covered in interagency ICDs prior to exchanging Pointing Request Messages on a regular basis.

ANNEX C provides a list of acronyms and abbreviations used in the Recommendation.

ANNEX D provides sample Pointing Request Messages.

ANNEX E specifies adopted attitude conventions.

ANNEX F provides d etails on the use of operators

ANNEX G lists supported units.

ANNEX H discusses security, SANA and pattent considerations for the Pointing Request Message.

1.6 DEFINITIONS

For the purposes of this document, the following definitions apply:

a) the words ‘shall’ and ‘must’ imply a binding and verifiable specification;

b) the word ‘should’ implies an optional, but desirable, specification;

c) the word ‘may’ implies an optional specification;

d) the words ‘is’, ‘are’, and ‘will’ imply statements of fact.

1.7 REFERENCES

The following documents contain provisions which, through reference in this text, constitute provisions of this Proposed Standard. At the time of publication, the editions indicated were valid. All documents are subject to revision, and users of this Proposed Standard are encouraged to investigate the possibility of applying the most recent editions of the documents indicated below. The CCSDS Secretariat maintains a register of currently valid CCSDS documents.

[1] *Time Code Formats*. Recommendation for Space Data System Standards, CCSDS 301.0-B-4. Blue Book. Issue 4. Washington, D.C.: CCSDS, November 2010.

[2] "JPL Solar System Dynamics." Solar System Dynamics Group. <http://ssd.jpl.nasa.gov/>

[3] IEEE Standard for Binary Floating-Point Arithmetic. IEEE Std 754-1985. New York: IEEE, 1985.

[4] Extensible Markup Language (XML) 1.0 (Fifth Edition) W3C Recommendation 26th November 2008 (http://www.w3.org/TR/xml/).

[5] XML Inclusions (XInclude) Version 1.0 (Second Edition) W3C Recommendation 15 November 2006 (http://www.w3.org/TR/xinclude/).

[6] Navigation Data Messages / XML Specification, CCSDS 505.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, December 2010.

[7] Orbit Data Message, CCSDS 502.0-B-2. Blue Book. Issue 2. Washington, D.C.: CCSDS, November 2009.

[8] Attitude Data Message, CCSDS 504.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, May 2008.

[9] http://naif.jpl.nasa.gov/pub/naif/toolkit_docs/FORTRAN/req/naif_ids.html (source for solar system bodies naming)

2 OVERVIEW

There are numerous circumstances in spacecraft operations, when pointing information has to be transmitted from a user, e.g. of an instrument or of a relay service to the operator of a spacecraft. For interagency operations, it is desirable to exchange information regarding these requested pointings in a standardized format.

All pointing requests have as a common, most basic element the specification of the attitude of an object or the direction of an axis defined relative to this object at an instant of time. The object, which defines a coordinate frame, can be a spacecraft, an instrument or sensor or an antenna mounted on a spacecraft or an articulated spacecraft component. It is possible to define the attitude relative to any known coordinate frame (e.g., an inertial frame or a rotating orbital frame) or the axis direction relative to another object (e.g., another spacecraft, a star, a solar system object or a feature on a solar system object).

The target may be an attitude relative to any defined coordinate frame: inertial coordinates, orbital coordinates, relative coordinates, etc. For partial attitudes the target direction may be to arbitrary vectors in the target frame, or to external directions defined by the positions of planets, other spacecraft, points on another object, etc. In all cases, an unambiguous method of linking the object coordinate system to the target must be available.

Pointing request messages can aggregate single pointing requests into time-dependent sequences such as raster scans.

The PRM will provide a vehicle to navigators, science teams and user/providers of relay services for the transmission of requested pointing sequences of varying complexity. Currently, this information is transmitted in common language or in various fixed file formats. Only recently a formal language representation is used for the transmission of science pointing requests for certain missions in ESA. Thus currently approaches differ for different missions even within the same space agency. The proposed standard seeks to offer an alternative to the various practices and formats currently in use.

2.1 POINTING REQUESTS IN SCIENCE OPERATIONS

Pointing requests are transmitted for instance from scientists who operate an onboard instrument to the operator of the respective S/C. These data transmissions could be inter-agency, for instance in case of projects which are done in collaboration between different agencies. Science pointing requests could be basic, e.g.: “point the boresight of an instrument for a given time period into an inertial direction or at an inertial target” but also more complex pointing requests commonly occur. Examples are:

- point the boresight of an instrument onboard a planetary orbiter at the limb of the illuminated section of the planet,
- point the onboard high gain antenna of a planetary orbiter at the earth such that the antenna beam passes the planet atmosphere at a given altitude,

- perform with the boresight of an instrument a raster scan of a target with a defined size, geometry, number of points and dwell time at each point.

2.2 POINTING REQUESTS IN RELAY OPERATIONS

The following are examples of pointing requests which are passed from the user of a relay service to the provider:

- point the relay antenna of spacecraft 1 (which serves as relay) to spacecraft 2 (which uses the relay service) during a given time period,
- point the relay antenna of a planetary orbiter to a lander or rover on the surface of the planet during a given time period,
- point the relay antenna of a planetary orbiter to a lander on approach to the planet while it passes through a given altitude range.

All above examples have occurred in practice in the context of cross-support between ESA and NASA missions at Mars.

2.3 IMPLEMENTATION BASICS

The PRM is implemented as an XML document only. The complexity of the pointing requests and the involved elements make necessary to provide an implementation that supports that complexity. XML is the natura manner of structuring the prointing requests in a flexble and extendable manner.

A prerequisite to understand, process and generate pointing request messages is to have sufficient knowledge in XML data representation and structuring. Knowledge in XML side technologies like Xpath, XSL and XML Schema are desirable but not stricitly necessary to understand the PRM principles.

The PRM is implemented as a collection of elements that define a hierarchical structure of data elements. One of the main priniples in th design of the PRM is the ability to create basic entities that can be aggregated into more complex strutures and operations. Besides it is possible to use reference mechanism that allow the systematic and consistent reuse of the defined data strcutures.

3 POINTING REQUEST MESSAGE

3.1 GENERAL

This section discusses the structure and content for the PRM.

Previously derived standards for exchange of navigation data, e.g. Orbit Data Message (ODM), Attitude Data Message (ADM) or Tracking Data Message (TDM), exist alternatively in Key-Value Notation (KVN) or XML representations. The PRM exists in XML notation only since the expected complexity of its structured data is not suitable for the KVN representation.

The PRM standard provides normative templates that cover common pointing scenarios (see section 4).

It is possible that there are mission specific pointing scenarios, which cannot be covered by any of the normative templates provided in this standard. In this case, mission specific PRMs can be developed based on the framework specified in the standard (see sections 3.2, 3.3 and 3.3.3.3) and recorded in the mission-specific ICD.

Section 5 provides the rules for the construction of a PRM from scratch using the general building elements in section 3.3.

3.2 PRM STRUCTURE

3.2.1 STRUCTURE OVERVIEW

3.2.1.1 The PRM shall consist of pointing request data pertaining to one spacecraft.

3.2.1.2 The PRM shall be structured in XML format.

3.2.1.3 The root element of a PRM shall be the `<prm>` element.

3.2.1.4 The standard NDM header as described in the NDM/XML (see [7], Section 4]) shall follow the `<prm>` tag.

3.2.1.5 The XML version, root element tag, and NDM/XML header shall be constructed as described in the NDM/XML [[7], Section 4].

3.2.1.6 The final attributes of the `<prm>` tag shall be 'id' and 'version'.

3.2.1.7 The '`id`' attribute shall be '`id="CCSDS_PRM_VERS"`'.

3.2.1.8 The '`version`' attribute for the `<prm>` shall be '`version="1.0"`'.

3.2.1.9 The `<prm>` element shall consist of two main parts; `<header>` and `<body>`.

3.2.1.10 The `<body>` element shall consist of a list of `<segment>` elements.

3.2.1.11 Each `<segment>` element shall consist of two main parts `<metadata>` and `<data>`, according to the following XML structure.

3.2.1.12 The `<metadata>` elements may contain either definitions (identified by `<definition>` elements) or definition references (identified by `<source>` elements).

3.2.1.13 The `<metadata>` element shall contain the `<TIME_SYSTEM>` child to define the reference time scale for the segment.

The following XML layout corresponds to the intended PRM structure according to the previous requirements.

```
<prm xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="ndmxml-1.0-master.xsd" id="CCSDS_PRM_VERS"
version="1.0">
	<header>
		<CREATION_DATE>2012-281T17:26:06</CREATION_DATE>
		<ORIGINATOR>ESA</ORIGINATOR>
	</header>
	<body>
		&lt;segment&gt;  <!-- Definition segment -->
			<metadata>
				<TIME_SYSTEM>UTC</TIME_SYSTEM>
				<definition name="xxx" version="1.5" />
				<definition name="defBlock" version="a.b" />
				<definition name="yyy" version="a.b" />
				<definition name="defBlock" version="1.5" />
				…
			</metadata>
			<data />  <!---  Empty (or absent) data block -->
		&lt;/segment&gt;
		&lt;segment&gt;  <!-- First pointing request -->
			<metadata>  <!-- Definitions for the first request -->
			            <!-- Not all necessarily referenced later -->
				<TIME_SYSTEM>UTC</TIME_SYSTEM>
				<definition name="zzz" version="1.0" />
				<source name="xxx" version="1.5" />
				<source name="yyy" version="a.b" />
			</metadata>
			<data>    <!-- Pointing request data for the first request -->
			</data>
		&lt;/segment&gt;
		&lt;segment&gt;   <!-- Second pointing request -->
			<metadata>  <!--    Definitions for the second request -->
				<TIME_SYSTEM>UTC</TIME_SYSTEM>
				<source name="xxx" version="a.b" />
			</metadata>
			<data>   <!-- Pointing request data for the second request -->
			</data>
		&lt;/segment&gt;
		.
		.
		.
		&lt;segment&gt;  <!-- n-th pointing request -->
			<metadata>    <!-- Definitions for the n-th request -->
				<TIME_SYSTEM>UTC</TIME_SYSTEM>
				<source name="xxx" version="1.5" />
```

```
                </metadata>
                <data>   <!-- Pointing request data for the n-th request -->
                </data>
            &lt;/segment&gt;
        </body>
</prm>
```

Figure 3-1: PRM Structure Example

3.2.1.14 If the PRM data is contained in a file, the file naming scheme should be agreed to on a case-by-case basis between the participating agencies and documented in an ICD.

3.2.1.15 If the PRM data is contained in a file, the method of exchanging files should be decided on a case-by-case basis by the participating agencies and documented in an ICD.

3.2.2 POINTING REQUEST ELEMENTS DEFINITIONS

3.2.2.1 The definitions shall be kept in separate data structures in the `<metadata>` section.

3.2.2.2 The root element of each definition shall be the `<definition>` element.

3.2.2.3 A `<prm>` element may include one or more definitions contained in the `metadata` section.

3.2.2.4 If the definitions are incorporated to the PRM by means of the `<include>` element, the resulting expanded PRM shall comply with the general NDM structure defined in 3.2.1.

3.2.2.5 The number of definitions and their content should be agreed to on a case-by-case basis between the participating agencies and documented in an ICD.

3.2.2.6 Definitions shall specify elements used in the `<prm>` body, e.g. alignments, boresight directions and directions to targets.

3.2.2.7 The elements defined in the definitions shall be referred to in the PRM body by name.

3.2.2.8 The definitions shall include exactly one *root frame*.

3.2.2.9 The definitions shall include one or more *independent frames* defined relative to the *root frame* or to another *independent frame.*

3.2.3 POINTING REQUEST BODY DESCRIPTION

3.2.3.1 The pointing request body shall contain one or more attitude timelines.

3.2.3.2 An attitude timeline shall describe the attitude of a spacecraft or any of its articulate parts over a period of time.

3.2.3.3 For each independent frame defined in the definitions there shall be one attitude timeline in the pointing request body.

3.2.3.4 An attitude timeline shall consist of a series of attitude blocks.

3.2.3.5 The root element of the attitude timeline shall be the `timeline` element.

3.2.3.6 The number of attitude timelines and their content should be agreed to on a case-by-case basis between the participating agencies and documented in an ICD.

3.2.3.7 For each attitude block the attitude shall be defined over a certain interval of time by means of the `block` element.

3.3 POINTING REQUEST ELEMENTS

3.3.1 POINTING REQUEST ELEMENTS OVERVIEW

3.3.1.1 The text data contained in XML elements shall be formatted according to the data types defined in Table 1.

3.3.1.2 Depending on the specific use case, some of the physical or mathematical entity types defined in Table 1 may not appear in a PRM.

3.3.1.3 The attributes and/or child elements or text contents of the XML elements defining the respective entity type shall be as defined in section 3.3.2.

3.3.1.4 All child elements and attributes which are not specified as obligatory shall be considered optional.

3.3.1.5 In addition to the specific attributes which are defined for each entity type, any element may contain the following optional attributes: `name`, `ref` and `localName`.

Note: These attributes fulfill special functions in the naming-referencing mechanism described in 3.4.

Table 1: Overview of entity types described by XML elements

Entity type	Generic element name	Type description
Integer	`integer`	Describes an integer number. An integer shall be dimensionless. Basic type is `xsd:integer`

Entity type	Generic element name	Type description
List of integers	`integerList`	Describes a list of integers separated by white space. All integers in a list shall be dimensionless. List of integers may have any length.
Real	`real`	Describes a real number. The real can be dimensionless or have a unit (allowed units are listed in ANNEX G). Basic type is `xsd:double`
List of reals	`reaList`	Describes a list of reals separated by white space. All reals in a list have the same units. Allowed units are listed in ANNEX G. List of reals may have any length
Epoch	`epoch`	Describes an instant in time. Epoch entities are used for instance to build timelines. Basic type is `ndm:epochType` (refer to [6])
List of epochs	`epochList`	Describes a list of instants in time (epochs) separated by white space. List of times may have any length.
Duration	`duration`	Describes an elapsed period of time. Duration entities are used to build epochs relative to other epochs. Basic type is `ndm:durationType` (refer to [6])
List of durations	`durationList`	Describes a list of elapsed times (durations) separated by white space. List of durations may have any length.
Direction vector	`dirVector`	Describes a direction vector (unit vector or right ascension and declination provided as a list of reals). Direction vectors are defined relative to a frame. When given as unit vector the contents are dimensionless.
State Vector	`stateVector`	Describes one orbital state defined as an epoch and the postion and velicity at that epoch in Cartesian coordinates. Basic type is `ndm:stateVectorType`

Entity type	Generic element name	Type description
Orbit entity	`orbit`	Describes a sequence of state vectors as a function of time. State vectors are used to model trajectories of objects relative to the root frame. An orbit entity may be given as the implicit ephemeris of a celestial object or the time varying position of a target point.
Surface	`surface`	Describes a surface. A surface can be described in different ways depending on its type, e.g. a sphere is defined by its centre and radius.
Surface vector	`surfaceVector`	Describes a trajectory over a surface.
Reference frame entity	`frame`	Describes a reference frame. Different types of reference frames can be defined (see ANNEX E).
Attitude block	`block`	Defines the attitude during a time interval.
Attitude	`attitude`	Describes the attitude provided as three coordinate axes that may be a function of time. Attitude entities are used to describe the orientation of a reference frame with respect to another.
Phase angle	`phaseAngle`	Rotation angle around a direction with respect to a zero reference. Describes a condition for solving a rotational degree of freedom in the orientation of a reference frame.
Angular rate	`angularRate`	Rotation rate around a direction. Describes the rotation condition for the cases when a rotation around an axis is undefined but the rotation rate is known
Rotation	`rotation`	Defines rotation to be applied to a direction vector or attitude.
String	`string`	Contains string data.

The tags provided in the Generic Element Name column indicates a default XML name for generic element use. In specific cases the value of the XML tag is defined by the context where it is defined and used being then of the type defined in the Entity Type column. The definition in the Entity Type column is the unique identifier for each given data type.

3.3.2 DETAILED DEFINITIONS OF POINTING REQUEST ELEMENTS

3.3.2.1 Epoch type

3.3.2.1.1 An instant in time shall be represented by an element of type ***Epoch***.

Representation	**Elements description**	**Example**
Epoch	Optional attribute `format` of default value `calendar` (allowed values: `calendar`, `DOY`). The text content format depends on the value of the `format` attribute. (see Table 1).	`<epoch>` `2000-01-01T00:00:00` `</epoch>` `<epoch format='DOY'>` `2000-001T00:00:00` `</epoch>`
Reference epoch plus duration	`refEpoch` child element of type ***Epoch.*** `duration` child element of type ***Duration*** and time type units.	`<epoch>` `<refEpoch …> 1` `</refEpoch> <duration` `units='dhms'>` `10:00.` `</duration>` `</epoch>`
Epoch from events file	`eventsFile`: the URL of the file containing the events that define the time series. `eventId`: the user defined identification of the event to be used for the definition of the timeline. `eventCount`: the occurrence of the event with `eventId` that defines the selected time from the time series.	`<epoch>` `<eventsFile …>` `<eventId …>` `<eventCount …>` `</epoch>`

3.3.2.2 List of epochs type

3.3.2.2.1 A list of instants in time shall be represented by an element of type ***List of Epochs***.

3.3.2.2.2 The epochs in a list of epochs shall be chronologically ordered.

3.3.2.2.3 The difference between two consecutive epochs in a list of epochs shall be greater than zero.

[1] The notation … is used for elements whose representation is partial to ease reading.

Representation	**Elements description**	**Example**
List of epochs	Optional attribute `format` of default value `calendar` (allowed values: `calendar`, `DOY`). The text content format depends on the value of the `format` attribute (see Table 1).	`<epochList>` `2008-07-10T00:00:00` `2008-07-10T01:00:00` `</epochList>` `<epochList format='DOY'>` `2008-071T00:00:00` `2008-071T01:00:00` `</epochList>`
Reference epoch plus list of durations	`refEpoch` child element of type ***Epoch***. `duration` element of type ***Duration*** and time type units. The resulting list is a list of absolute epochs with the same number of components as the `durationList` entity. Each epoch in the resulting list is the result of adding each duration from `durationList` to the reference time defined by the `refEpoch` element. All durations in `durationList` shall be in the same time scale as the epoch in `refEpoch`.	`<epochList>` `<refEpoch …>` `<durationList …>` `</epochList>`

3.3.2.3 Duration type

3.3.2.3.1 An elapsed period of time shall be represented by an element of type duration.

Representation	**Elements description**	**Example**
Duration	Duration.	`<duration>03:00:00</duration>`

3.3.2.4 List of durations type

3.3.2.4.1 A list of elapsed periods of time shall be represented by an element of type list of durations.

Representation	**Elements description**	**Example**
List of durations	List of durations	`<durationList>` `00:02:00 00:00:10` `00:02:00 00:00:20` `</durationList>`

3.3.2.5 Integer type

3.3.2.5.1 An integer number shall be represented by an element of type ***Integer***.

Representation	Elements description	Example
Integer value	Text contents of data type ***Integer***.	<integer>1</integer>
Integer operator	operator attribute identifying the operation to be performed (allowed values: plus, minus, multiply; not allowed values (incomplete list): division). Two or more Integer child elements of type ***Integer***.	<integer operator='plus'> <integer>1</integer> <integer>2</integer> </integer>

3.3.2.6 List of integers type

3.3.2.6.1 A list of integers shall be represented by an element of type ***List of Integers***.

Representation	Elements description	Example
List of integers	Text contents of data type ***List of Integers.***	<integerList> 1 2 3 </integerList>
Integer list operation.	operator attribute identifying the operation to be performed plus the child elements over which the operation is performed. See description of allowed list operators and child elements in ANNEX F.	<integerList operator='plus'> <integerList …> <integerList …> </integerList>

3.3.2.7 Real type

3.3.2.7.1 A real shall be represented by an element of type ***Real***.

Representation	Elements description	Example
Real value	Text contents of data type ***Real***. Optional attribute units (see allowed values in ANNEX G).	<real units="m">1.2</real>
Real operator	operator attribute identifying the operation to be performed (allowed values: plus, minus, multiply, divide, unaryMinus). Two or more real elements of type ***Real***. Restrictions to units apply for certain operators (see ANNEX F).	<real operator='plus'> <real>0.1</real> <real>0.2</real> </real>

Representation	Elements description	Example
Interpolation table	Child elements: `epochList` of type ***List of Epochs***, `valueList` of type ***List of Reals***. `derivativeList` of type ***List of Reals*** is optional. All lists shall have the same length. The units of `derivativeList` shall match the type of dividing the units of `valueList` by units of time. This representation describes an interpolation table. The algorithm to be used depends on the implemented data exchange and shall be documented in the appropriate ICD. The `derivativeList` is optional for those algorithms interpolating simultaneously the values and their derivatives (e.g. splines).	`<real>` `<epochList …>` `<valueList …>` `<derivativeList …>` `</real>`

3.3.2.8 List of reals type

3.3.2.8.1 A list of reals shall be represented by an element of type list of reals.

Representation	Elements description	Example
List of reals	Text contents of data type ***List of Reals***. Optional `units` attribute (see ANNEX G).	`<realList>` `1. 2. 3.` `</realList>`
Real list operation.	`operator` attribute identifying the operation to be performed plus the child elements over which the operation is performed. See description of operators and child elements in ANNEX F.	`<realList operator='plus'>` `<realList …>` `<realList …>` `</realList>`

3.3.2.9 Direction vector type

3.3.2.9.1 A direction vector shall be represented by an element of type ***Direction Vector***.

3.3.2.9.2 Each direction vector is defined relative to a frame (see 3.3.2.14).

Representation	**Elements description**	**Example**
Coordinates	Optional attribute `type` (default value is `cartesian`). Allowed values for `type`: `cartesian` (for which the text content is a list of 3 real numbers), `spherical` (for which the text content is a list of 2 real numbers) `raDec` (for which the text content is a list of 2 real number representing right ascension and declination). Obligatory attribute `frame` of string type. The value of the `frame` attribute shall be equal to the name of one of the `frame` elements defined in ANNEX A. Optional `units` attribute of angle units type if the value of `type` is `spherical` or `raDec`. For the allowed values of the `units` attribute see ANNEX G. If Cartesian coordinates are provided, the direction vector defined results from the normalization of the coordinates. This representation represents a fixed direction vector.	`<dirVector frame=’SC’>` `0. 0. 1.` `</dirVector>`
Origin plus Target trajectory	`origin` and `target` child elements of ***Orbit entity*** type. The direction vector described is the result of normalizing the vector from the trajectory defined by the `origin` element to the trajectory defined by the `target` element.	`<dirVector frame=’EME2000’>` `<origin …>` `<target …>` `</dirVector>`
Rotated direction vector	Child element `dirVector` of ***Direction vector*** type plus `rotation` child element of ***Rotation*** type. The resulting direction vector is defined relative to the same frame as the child `dirVector` element.	`<dirVector>` `<dirVector …>` `<rotation …>` `</dirVector>`
Direction at epoch	`dirVector` element of ***Direction vector*** type plus `refEpoch` element of ***Epoch*** type. The resulting direction vector is the direction vector corresponding to the value of the direction vector defined by the `dirVector` child element at the epoch defined by the `refEpoch` child element.	`<dirVector>` `<dirVector …>` `<refEpoch …>` `</dirVector>`
Direction vector operation	`operator` attribute of data type ***String***. Allowed values are: `cross`, `derivative, unaryMinus`. `dirVector` The child elements are of ***Direction vector*** type.	`<dirVector operator="cross">` `<dirVector …>` `<dirVector …>` `</dirVector>`

Representation	Elements description	Example
	The second child element is optional and will not be provided if `operator` value is `derivative` or `unaryMinus`. The frames of both direction vectors must be defined relative to the same independent frame or root frame (see 3.3.2.14).	
Surface direction	`surfaceVector` element of type ***Surface vector*.** `operator` attribute of data type ***String***. Allowed values are: `tangent`, `normal`. The operator tangent can only be applied if the surface vectors as function of time in the frame in which the surface is defined has a non-zero time derivative. The tangent points in the direction of that derivative in direction of ascending time.	<dirVector operator="normal"> <surfaceVector …> </dirVector>

3.3.2.10 State Vector type

3.3.2.10.1 An orbital state shall be represented by an element of type ***State Vector***.

Representation	Elements description	Example
State vector	Text contents of data type ***State Vector***. Contents is an instant in time of type ***Epoch*** and the contents of the state vector as defined in [6]. Optional attribute `units` (see allowed values in ANNEX G).	<stateVector> <epoch ...> <X ...> <Y ...> <Z ...> <XDOT ...> <YDOT ...> <ZDOT ...> </stateVector>

3.3.2.11 Orbit entity type

3.3.2.11.1 The orbit entity type shall be used to describe the position of an object versus time with respect to a base frame (see 3.3.2.14).

Representation	Elements description	Example
Ephemerides object.	`ephObject` element of data type ***String*** specifying the object name contained in the ephemeris according to ref [9] as default. Deviation from this naming convention shall be documented in an applicable ICD.	<orbit> <ephObject> MARS </ephObject> </orbit>

Representation	Elements description	Example
Orbit file	One `orbitFile` element of type ***String***. The `orbitFile` element contains the URL to the Orbit Ephemeris Message (OEM) containing the ephemeris of the object	`<orbit>` `  <orbitFile …>` `</orbit>`
Surface vector	One `surfaceVector` element of ***Surface vector*** type. The trajectory provided in this representation is a single point on the surface defined from any of the representations of the ***Surface Vector*** type.	`<orbit>` `  <surfaceVector …>` `</orbit>`

3.3.2.12 Surface

3.3.2.12.1 The surface type shall be used to describe reference surfaces in a frame dependent on the root frame (see 3.3.2.14). All represented surfaces are differentiable and convex.

Representation	Elements description	Example
Sphere	Obligatory attribute `frame` of ***String*** type. The value of the `frame` attribute shall be equal to the name of one of the `frame` elements defined in 3.3.2.14. `radius` element of type real with unit type distance. It shall define a constant real. `origin` element of type ***Orbit entity***.	`<surface frame='ITRF'>` `  <radius …>` `  <origin …>` `</surface>`
Ellipsoid	Obligatory attribute `frame` of ***String*** type. The value of the `frame` attribute shall be equal to the name of one of the frame elements defined in the PRM definition sections. Elements `a`, `b` and optionally `c` are of type real with unit type distance. `origin` element of type ***Orbit entity***.	`<surface frame='ITRF'>` `  <a …>` `  <b …>` `  <c …>` `  <origin …>` `</surface>`

3.3.2.13 Surface vector

3.3.2.13.1 The surface vector type shall be used to describe reference trajectories over surfaces with respect to a base frame (see 3.3.2.14).

Representation	Elements description	Example
Coordinates	`surface` element of type ***Surface***.	`<surfaceVector>` `  <surface …>`

Representation	Elements description	Example
	surfaceCoord element of type ***List of Reals*** with angle units defining the longitude and latitude of the point on the surface. The longitude and latitude are with respect to the origin of coordinates on the surface and in the frame of the surface. height element of type ***Real*** with distance units. The trajectory is defined by applying the height along the local surface normal of the point on the surface described by the previous elements.	<surfaceCoord …> <height …> </surfaceVector>
Surface normal from origin	surface element of type ***Surface***. origin element of type ***Orbit***. operator attribute of type ***String*** of fixed value normal. The trajectory described with this representation results in the point on the surface whose local normal direction points towards origin.	<surfaceVector operator='normal'> <surface …> <origin …> </surfaceVector>
Limb point from origin and target direction	surface element of type ***Surface***. origin element of type ***Orbit***. targetDir attribute of type ***Direction vector***. The trajectory described with this representation is the direction to a point on the limb seen from the origin. The point on the limb is the one defined by the intersection of the surface and the half-plane defined by the line connecting the surface origin and the origin with the positive component along targetDir. targetDir and the line connecting the surface origin and origin must not be aligned.	<surfaceVector> <surface …> <origin …> <targetDir …> </surfaceVector>

3.3.2.14 Reference frame entity type

3.3.2.14.1 The reference frame type shall be used to assign names and to describe the hierarchy of the reference frames used in the PRM.

Representation	Elements description	Example
Root frame	Obligatory name attribute of type string. Fixed value baseFrame attribute with value none. Only one root frame element is allowed.	<frameEntity name='EME2000' baseFrame='none' />

Representation	Elements description	Example
	See reference frames description in ANNEX A.	
Independent frame	Obligatory `name` and `baseFrame` attributes of type string. The `baseFrame` attribute shall correspond to the `name` of a previously defined `frame`. See reference frames description in ANNEX A.	`<frameEntity name='SC' baseFrame='EME2000' />`

3.3.2.15 Attitude type

3.3.2.15.1 An attitude type element shall always be a descendant of an attitude timeline or reference frame type.

3.3.2.15.2 The direction vectors corresponding to the `frameDir` and `baseFrameDir` element shall be defined relative to the respective frames of the corresponding attitude timeline or reference frame.

Representation	Elements description	Example
Directions	`frameDir` and `baseFrameDir` elements of type ***Direction vector***. `phaseAngle` element of ***Phase Angle*** type. See attitude description in ANNEX E.	`<attitude>` `<frameDir …>` `<baseFrameDir …>` `<phaseAngle …>` `</attitude>`
Rotated attitude.	`attitude` element of ***Attitude*** type (optional). Element `rotation` of ***Rotation Entity*** type. See rotated attitude description in ANNEX E.	`<attitude>` `<attitude …>` `<rotation …>` `</attitude>`

3.3.2.15.3 The direction vectors corresponding to the `frameDir` and `baseFrameDir` element shall be defined relative to the respective frames of the corresponding attitude timeline or reference frame.

3.3.2.16 Attitude block type

3.3.2.16.1 The attitude block type shall be used to define the attitude of the independent frames (see 3.3.2.14).

Representation	Elements description	Example
Attitude function.	`startEpoch` and `endEpoch` elements of type ***Epoch***, `attitude` element of type ***Attitude***.	`<block>` `  <startEpoch …>` `  <endEpoch …>` `  <attitude …>` `</block>`

3.3.2.17 Phase Angle type

3.3.2.17.1 The `phaseAngle` element shall be a child element of an attitude type.

3.3.2.17.2 The directions corresponding to the `frameDir` and `baseFrameDir` elements must be defined relative to the respective frames of the parent attitude type element.

3.3.2.17.3 For the directions in the roll type element and attitude type parent element the following constraints apply:

(a) The two `frameDir` elements (the child of the `attitude` element and the child of `phaseAngle`) shall not result in two parallel directions for the time interval where the attitude is to be described, since this would result in a not defined attitude.

(b) The two `baseFrameDir` elements (the child of the `attitude` element and the child of `phaseAngle`) shall not result in two parallel directions for the time interval where the attitude is to be described, since this would result in a not defined attitude.

Representation	Elements description	Example
Two directions kept at a certain angle	`frameDir` and `baseFrameDir` elements of type ***Direction vector*** plus `angle` element of type ***Angle***. See roll elements description in ANNEX E.	`<phaseAngle>` `  <frameDir …>` `  <baseFrameDir  …>` `  <angle …>` `</phaseAngle>`
Value for rotational degree of freedom	`frameDir` and `baseFrameDir` elements of type ***Direction vector*** plus `projAngle` element of type ***Angle***. See roll elements description in ANNEX E.	`<phaseAngle>` `  <frameDir …>` `  <baseFrameDir  …>` `  <projAngle …>` `</phaseAngle>`

3.3.2.18 Angular rate type

3.3.2.18.1 The `angularRate` element shall be a child element of an attitude type.

Representation	Elements description	Example
Angular velocity	Optional attribute `units` (see allowed values in ANNEX G).	`<angularRate units=”deg/s”>0.34 </angularRate>`

3.3.2.19 Rotation type

3.3.2.19.1 The rotation entity type shall always be a child element of an attitude type element or a direction vector type element.

Representation	Elements description	Example
Quaternion	`quaternion` of type `ndm:quaternionType` as defined in [6].	`<rotation scalar=’last’>` `<quaternion>` `<Q1 …>` `<Q2 …>` `<Q3 …>` `<QC …>` `</quaternion>` `</rotation>`
Rotation axis plus rotation angle	`axis` element of type ***Direction vector*** plus `angle` element of type ***Angle***. The rotation element defines a simple rotation (from a rotation axis and a rotation angle) to be applied to certain direction vector(s). The direction vector(s) to be rotated are defined by elements located at the same level in the tree as the `rotation` element. If the rotation type element is a child of an attitude type element then the direction vector corresponding to the `axis` is defined relative to the `baseFrame` or `frame` of the `attitude` element. If the rotation type element is a child of a direction type then the direction vector corresponding to the `axis` is defined relative to the same frame the parent direction vector is defined.	`<rotation>` `<axis …>` `<angle …>` `</rotation>`
Standard frame transformation	This representation describes transformations between standard frames. `from` and `to` attributes of string type.	`<rotation from=’EME2000’ to=’ITRF2000’ />`

Representation	Elements description	Example
Sequence of rotations	Several `rotation` elements of ***Rotation Entity*** type. The order of the `rotation` elements determines the order of application of the rotations.	`<rotation>` `  <rotation …>` `  <rotation …>` `</rotation>`

3.3.2.20 String

3.3.2.20.1 The string type shall be used to describe string data.

Representation	Elements description	Example
String	Text contents of data type string.	`<string> Example </string>`

3.3.3 AUXILIARY ELEMENTS

3.3.3.1 Include element

3.3.3.1.1 The `include` element shall be used incorporate a definition file into the PRM (see [5]).

3.3.3.1.2 The `include` element shall always be a child of the `<prm>` element.

Element description	Example
Attribute `href` of data type string that contains the filename of the file referenced (paths may be relative or absolute).	`<include` `  href='Definitions1.xml' />`

3.3.3.2 Definition element

3.3.3.2.1 The `definition` element shall be used to group a list of definitions (named entities).

3.3.3.2.2 The `definition` element shall always be the root of a definition file.

Element description	Example
List of elements of any entity type as described in Table 1 in any number. The generic element name corresponding to the type (see Table 1) shall be used.	`<definition>` `  <real name='one'> 1. </real>` `  <real name='two'> 2. </real>` `</definition>`

3.3.3.3 Timeline element

3.3.3.3.1 The Timeline element shall be used to define the attitude of an *independent frame* relative to a base frame (either the *root frame* or another *independent frame*).

3.3.3.3.2 The `timeline` element shall always be a child of the PRM element.

3.3.3.3.3 The `timeline` element shall be composed of `block` elements sorted in chronological order.

Element description	**Example**
Sequence of one or more `block` elements of type ***Attitude block***. The value of the `frame` attribute identifies one of the independent fames previously defined.	`<timeline frame='SC'>` `  <block …>` `  <block …>` `  <block …>` `</timeline>`

3.4 THE NAMING AND REFERENCING MECHANISM

3.4.1 NAME ASSIGNATION

3.4.1.1 Any element of the types defined in 0 that is a child of a `definition` element shall include a `name` attribute to identify the element.

3.4.1.2 Any element of the types defined in 0 inside the PRM file body may include a `name` attribute to identify the element.

3.4.1.3 The value of the `name` attribute of an element shall be unique among the entity type of the element and considering both the PRM body plus all definitions.

3.4.2 NAME REFERENCING

3.4.2.1 Any element of the types defined in 3.3.2 inside the PRM body or definition may include a `ref` attribute to refer to another element by its `name` attribute.

3.4.2.2 The value of the `ref` attribute of an element shall match the value of the `name` attribute of one of the elements of the same element type (as defined in 3.3.2) that appears before in the PRM body or definition.

3.4.2.3 Any element containing the `ref` attribute shall be designated as *referencing element* and the element with the same value in the `name` attribute as *referenced element*.

3.4.2.4 A referencing element shall not be a descendant of the corresponding referenced element.

3.4.2.5 The referencing element shall not follow the element type content as defined in 3.3.2. Their allowed child elements are given by the parameters of the referenced element, as defined in 3.4.3.

3.4.3 DEFINING, USING AND OVERRIDING PARAMETERS

3.4.3.1 General

The use of parameters is intended to allow deferred instantiation of PRM elements between the definition and the body of the PRM request. This use case corresponds to the situation in which the information about the pointing element cannot be fully defined before the pointing timeline is completed. The parameter mechanism allows that an element that is described in the definition section (e.g. the axis of the instrument to be pointed) can be further referenced and completed in the body section within a timeline to define the direction to point to (e.g. the direction towards the instrument axis has to point).

The following requirements define the implementation of parameters within a parent element. The terminology used refers to the parent as the element containing the parameters and children as all elements within the parent that may be parameters of regular elements within the parent.

The referenced parent element declares some or all its children to be parameters by assigning a local name to them. The referencing parent element generates one child for each parameter in the reference parent element such that the local names are used to generate children elements within the referencing parent.

3.4.3.2 A parent element that defines a parameter construct shall have the `name` attribute.

3.4.3.3 The `localName` attribute shall be used to identify the children of a parent element that are parameters.

3.4.3.4 The name of every parameter shall be unique within the parent element.

3.4.3.5 Only strings that result in valid XML element names (See ref [4]) shall be used as the value for the `localName` attribute.

3.4.3.6 An element with the `localName` attribute shall only act as the parameter of the parent and not as the parameter of any ancestor of the parent.

3.4.3.7 A parent referencing element of a parameter construct shall have the `ref` attribute.

3.4.3.8 If the referenced parent element does not have parameters, the referencing parent element shall be an empty element.

Note: Parent elements that contain only regular children and no parameters do not expand any children in the parent referencing process. Regular elements are fully defined in the declaration of the parameter construct (within the referenced parameter) and the resulting referencing parent is therefore an empty element.

3.4.3.9 If the referenced parent element has parameters, the referencing parent element shall define child elements for all parameters in the referenced parent element.

Note: The child element name and type is given by the parameter name (i.e. value of `localName`*) and type (type of the child element in the referenced parent).*

3.4.3.10 The referencing *aren't element shall be built substituting each parameter in the referenced parent element with the corresponding child element in the referencing prent element.

Note: If the referencing element contains child elements corresponding to the referenced element parameters, the entities described by the referencing and the referenced element differ.

3.4.3.11 When a referenced element is descendant of a `definition` element, the parameter elements may be left empty.

3.4.3.12 When a parameter of a `definition` element is left empty then it shall be present as a child of the referencing element,

Note: a parameter may be given a value not requiring then further substitution in the referencing parent element; in this case the value is that of the parameter within the referenced parent element. This can be interpreted as a default value for the parameter. When the parameter is given no value within the referenced pared element then it is necessary to expand it in the referencing parent element.

3.4.4 EXAMPLES

The following example shows the naming of elements and element parameters and default substituion.

```
<dirVector name="axis1">
  <dirVector frame="EME2000" localName="Parameter1"> 0. 0. 1. </dirVector>
  <rotation>
    <!--- Naming of an element to be Tree2 --->
    <rotation name="rotation1">
      <axis frame="EME2000"> 1. 0. 0. </axis>
      <!--- Naming of a parameter to be angle1 --->
      <!--- Parameter has default units and value --->
      <angle localName="angle1" units="deg"> 0. </angle>
    </rotation>
```

```
    <rotation name="rotation2">
      <axis frame=" EME2000"> 0. 1. 0. </axis>
      <!--- Naming of a parameter to be Parameter3 --->
      <!--- Parameter has de fault units but no default value --->
      <angle localName="angle2" units="deg" />
    </rotation>
  </rotation>
</dirVector>
```

Referencing and parameter substitution (conventional substitution)

```
<!--- Conventional subsitution where units are taken by default from
      referenced parameters and values are given --->
<block ref="rotation1">
     <angle1>180.0</angle1>
</block>
<block ref="rotation2">
     <angle2>90.0</angle2>
</block>
```

Referencing and parameter substitution (with units overriding in substitution)

```
<!--- Subsitution where units are overriden and values are given --->
<block ref="rotation1">
     <angle1 units="rad">3.141593</angle1>
</block>
<block ref="rotation2">
     <angle2>90.0</angle2>
</block>
```

Referencing and parameter substitution (all by default)

```
<!--- All subsitution by default from reference parameters --->
<!--- Rotation 1 angle is 0.0 in degrees taken from referenced element --->
<block ref="rotation1" />
<!--- Angle in rotation 2 cannot be given by default as it has no value
      given in the referenced parent element and therefore mut be present --->
<block ref="rotation2">
     <angle2>90.0</angle2>
</block>
```

4 PRM TEMPLATES FOR COMMON, GENERIC POINTING SCENARIOS

4.1 GENERAL

4.1.1 If a pointing request inside a PRM can be represented by one of the pointing requests listed in this section, then the corresponding templates shall be used to build the corresponding PRM definitions and pointing request blocks.

4.1.2 If more than a template from this section is used inside a PRM, the used templates shall be combined following the rules from Section 3.

4.1.3 The example values provided for the variables in the PRM templates (between % symbols) shall be substituted by the proper values following the rules from Section 3 (a dash "-" character in the ‘Allowed values’ column indicates no restriction on allowed values other than that associated with the data type).

4.2 INERTIAL POINTING

4.2.1 The inertial pointing templates in this section shall be used to define a SC pointing request that fulfills the following conditions:

- a SC axis is pointed towards an inertial target,
- the remaining degree of freedom in the SC attitude is determined by a phase angle from a reference inertial direction to another SC axis,
- the SC axis and reference inertial direction used to define the phase shall not be parallel to the SC pointed axis and target direction respectively,
- the phase angle is the angle in the plane perpendicular to the target direction from the projection of the reference inertial direction to the projection of the SC axis, a positive angle meaning a positive rotation around the target direction. The resulting SC attitude is defined in ANNEX E,
- the offset angle is the angle around an arbitrary direction defined by the user to move away from the selected inertial target. The resulting SC attitude is defined in ANNEX E.

4.2.2 DEFINITION FILE TEMPLATE

4.2.2.1 The following template shall be used to build the definitions for a PRM containing inertial pointing requests. The variable content is shown by variable names between % symbols.

```
<metadata>
  <TIME_SYSTEM>UTC</TIME_SYSTEM>
  <definition name="%definitionName%" version="%definitionVersion%" />
    <frame baseFrame="none" name="%inertialFrameName%" />
```

```
    <frame baseFrame="%inertialFrameName%" name="%spacecraftFrameName%" />
    <block name="inertial">
      <startEpoch localName="blockStart" />
      <endEpoch localName="blockEnd" />
      <attitude>
        <frameDir localName="boresight" />
        <baseFrameDir localName="target" />
        <!-- Phase angle provides the rotation around the boresight -->
        <!-- For spin stabilized spacecraft omit this block -->
        <phaseAngle>
          <!-- SC reference direction for phase angle -->
          <frameDir frame="%spacecraftFrameName%"
                    coord="%phaseCoordType%"
                    units="%phaseFrameUnits%">%phaseCoords%</frameDir>
          <!-- Inertial reference direction for phase angle -->
          <baseFrameDir frame="%inertialFrameName%"
                        coord="%phaseBaseCoordType%"
                        units="%phaseBaseFrameUnits%"> %phaseBaseCoord% </baseFrameDir>
          <projAngle localName="phaseAngle" />
        </phaseAngle>
        <!-- Offset with respect to the boresight -->
        <!-- Block optional; remove if no offset with respect to target -->
        <offsetAngle>
          <!-- SC reference direction for offset angle -->
          <frameDir frame="%spacecraftFrameName%"
                    coord="%offsetCoordType%"
                    units="%offsetFrameUnits%">%offsetCoords%</frameDir>
          <!-- Inertial reference direction for offset angle -->
          <baseFrameDir frame="%inertialFrameName%"
                        coord="%offsetBaseCoordType%"
                        units="%offsetBaseFrameUnits%">%offsetBaseCoord%</baseFrameDir>
          <projAngle localName="offsetAngle" />
        </offsetAngle>
      </attitude>
    </block>
  </definition>
</metadata>
```

4.2.2.2 The variable content in the definitions template shall be substituted according to the rules in the following table. The values provided in the Tag column are those in the container: `/prm/body/segment/metadata/definition/`

Variable	Tag	Description	Allowed values	Example value
`%definitionName%`	`@name`	The identifier for the pointing elements definition; to be referenced in the generation of requests	-	
`%definitionVersion%`	`@version`	Version of the definition	By convention	`1.3`
`%inertialFrameName%`	`frame[1]/@name`	Inertial reference frame name.	One of the inertial frames from ANNEX A.	`EME2000`

Variable	Tag	Description	Allowed values	Example value
	`frame[2]/@baseframe`			
`%spaceraftFrameName%`	`frame[2]/@name`	SC reference frame name	-	`SC`
`%phaseBaseCoordType%`	`block/attitude/phaseAngle/baseFrameDir/@coord`	Type of coordinates defining the direction of the phase direction vector in inertial frame.	cartesian spherical	`cartesian`
`%phaseBaseFrameUnits%`	`block/attitude/phaseAngle/baseFrameDir/@units`	Units of the phase direction vector in inertial reference frame	For `%phaseBaseCoordType%=spherical:` `units="deg"` or `units="rad"` For `%phaseBaseCoordType%=cartesian` this variable must be an empty string.	`deg`
`%phaseBaseCoords%`	`block/attitude/phaseAngle/baseFrameDir`	The value of the direction vector coordinates to be used as reference for the computation of the phase angle in inertial frame.	Any conforming to the direction type in Table 1	`0. 0. 1.`
`%phaseCoordType%`	`block/attitude/phaseAngle/frameDir/@coord`	Type of coordinates defining the direction of the phase direction vector in SC frame.	`cartesian` `spherical`	`cartesian`
`%phaseFrameUnits%`	`block/attitude/phaseAngle/frameDir/@units`	Units of the phase direction vector in SC reference frame	For `%phaseCoordType%=spherical:` `units="deg"` or `units="rad"` For `%phaseCoordType%=cartesian` this variable must be an empty string.	`deg`
`%phaseCoords%`	`block/attitude/phaseAngle/frameDir`	The value of the direction vector coordinates in SC frame to compute the phase	Any conforming to the direction type in Table 1	`0. 1. 0.`

Variable	Tag	Description	Allowed values	Example value
		angle with respect to the base phase coordinates		
%offsetBaseCoordType%	block/attitude/offsetAngle/baseFrameDir/@coord	Type of coordinates defining the direction of the offset direction vector in inertial frame.	cartesian spherical	cartesian
%offsetBaseFrameUnits%	block/attitude/offsetAngle/baseFrameDir/@units	Units of the offset direction vector in inertial reference frame	For %offsetBaseCoordType%=spherical: units="deg" or units="rad" For %offsetBaseCoordType%=cartesian this variable must be an empty string.	deg
%offsetBaseCoords%	block/attitude/offsetAngle/baseFrameDir	The value of the direction vector coordinates to be used as reference for the computation of the offset angle in inertial frame.	Any conforming to the direction type in Table 1	0. 0. 1.
%offsetCoordType%	block/attitude/offsetAngle/frameDir/@coord	Type of coordinates defining the direction of the offset direction vector in SC frame.	cartesian spherical	cartesian
%offsetFrameUnits%	block/attitude/offsetAngle/frameDir/@units	Units of the offset direction vector in SC reference frame	For %offsetCoordType%=spherical: units="deg" or units="rad" For %offsetCoordType%=cartesian this variable must be an empty string.	deg
%offsetCoords%	block/attitude/offsetAngle/frameDir	The value of the direction vector coordinates in SC frame to compute the offset angle with respect to the base offset coordinates	Any conforming to the direction type in Table 1	0. 1. 0.

4.2.2.3 The direction vector type variables (Phase inertial reference direction and Phase SC reference direction) shall be given by its coordinates following the coordinates representation for direction vector type from section 3.3.2.9.

4.2.3 REQUEST BODY TEMPLATE

4.2.3.1 The following template shall be used to build inertial pointing request blocks inside the PRM body. The variable content is shown between % symbols.

```
<data>
  <timeline frame="%spacecraftFrameName%">
    <block ref="inertial">
      <!-- Pointing request start time -->
      <blockStart>%blockStartEpoch%</blockStart>
      <!-- Pointing request end time -->
      <blockEnd>%blockEndEpoch%</blockEnd>
      <!-- SC axis to be pointed to Target -->
      <boresight frame="%spacecraftFrameName%"
                 coord="%spacecraftCoordType%"
                 units="%spacecraftFrameUnits%">%spacecraftAxisCoords%</boresight>
      <!-- Inertial Target -->
      <target frame="%inertialFrameName%"
              coord="%inertialFrameCoordType%"
              units="%inertialFrameUnits%">%inertialFrameCoords%</target>
      <!-- Roll angle, see convention in Annex E  -->
      <!-- For spin stabilized spacecraft omit this block -->
      <phaseAngle units="%phaseAngleUnits%">%phaseAngle%</phaseAngle>
      <!-- Offset angle, see convention in Annex E  -->
      <!-- Block optional; remove if no offset with respect to target -->
      <offsetAngle units="%offsetAngleUnits%">%offsetAngle%</offsetAngle>
    </block>
  </timeline>
</data>
```

4.2.3.2 The variable content in the pointing request block template shall be substituted according to the rules in the following table. The values provided in the Tag column are those in the container: `/prm/body/segment/data/timeline/block/`

Variable	Tag	Description	Allowed values	Example value
`%spacecraftFrameName%`	`../@frame` `boresight/@frame`	SC reference frame name	-	`SC`
`%blockStartEpoch%`	`blockStart`	Start epoch of the pointing request	Epoch according to 3.3.2.1	`2009-09-25T19:00:00.`
`%blockEndEpoch%`	`blockEnd`	End epoch of the pointing request	Epoch according to 3.3.2.1	`2009-09-25T20:00:00.`
`%spacecraftCoordType%`	`boresight/@coord`	Coordinate type of the given pointed axis	`cartesian` `spherical`	`cartesian`
`%spacecraftFrameUnits%`	`boresight/@units`	Units of the direction vector in SC reference frame	For `%spacecraftCoordType%=spherical:`	`deg`

Variable	Tag	Description	Allowed values	Example value
			`units="deg"` or `units="rad"` For `%spacecraftCoordType%=cartesian` this variable must be an empty string.	
`%spacecraftAxisCoords%`	`boresight`	Coordinates of the direction vector in the SC reference frame	Any conforming to the direction type in Table 1	`0.052336 0. 0.99863`
`%inertialFrameName%`	`target/@frame`	Inertial reference frame name	One of the inertial frames from ANNEX A.	`EME2000`
`%inertialFrameCoordType%`	`target/@coord`	Type of the direction vector	`cartesian` `spherical`	`spherical`
`%inertialFrameUnits%`	`target/@units`	Units of the direction vector in inertial reference frame	For `%inertialFrameCoordType%=spherical:` `units="deg"` or `units="rad"` For `%inertialFrameCoordType%=cartesian` this variable must be an empty string.	`deg`
`%inertialFrameCoords%`	`target`	Coordinates of the direction vector in the inertial reference frame	Any conforming to the direction type in Table 1	`279.235 38.784`
`%phaseAngleUnits%`	`phaseAngle/@units`	Units for the phase angle	`deg` `rad`	`Deg`
`%phaseAngle%`	`phaseAngle`	The phase angle around the reference direction.	Angle value according to the real value representation in 3.3.2.6	`10.`
`%offsetAngleUnits%`	`offsetAngle/@units`	Units for the offset angle	`deg` `rad`	`Deg`
`%offsetAngle%`	`offsetAngle`	The anular offset applied with respect to the reference direction.	Angle value according to the real value representation in 3.3.2.6	`10.`

4.2.3.3 The values for the inertial reference frame and SC reference frame names shall match the definitions.

4.2.3.4 The direction vector type variables (boresight and target direction) shall be given by its coordinates following the coordinates representation for direction vector type from section 3.3.2.9.

4.3 SUN POINTING

4.3.1 The Sun pointing template in this section shall be used to define a SC pointing request that fulfills the following conditions:

- a SC axis is pointed towards the direction of the Sun,
- the rotation around the SC pointed axis is left free and a rotation rate may be provided.

4.3.2 DEFINITION FILE TEMPLATE

4.3.2.1 The following template shall be used to build the definitions for a PRM containing Sun pointing requests. The variable content is shown by variable names between % symbols.

```
<metadata>
  <TIME_SYSTEM>UTC</TIME_SYSTEM>
  <definition name="%definitionName%" version="%definitionVersion%" />
    <frame baseFrame="none" name="%inertialFrameName%" />
    <frame baseFrame="%inertialFrameName%" name="%spacecraftFrameName%" />
    <frame baseFrame="none" name="%spacecraftFrameName %" />
    <block name="sunPointing">
      <startEpoch localName="blockStart" />
      <endEpoch localName="blockEnd" />
      <attitude>
        <frameDir localName="boresight" />
        <baseFrameDir>
          <origin>
            <orbitFile>%OEM%</orbitFile>
          </origin>
          <target>
            <orbit name="Sun">
              <ephObject>SUN</ephObject>
            </orbit>
          </target>
        </baseFrameDir>
        <!-- Offset with respect to the boresight -->
        <!-- Block optional; remove if no offset with respect to target -->
        <offsetAngle>
          <!-- SC reference direction for offset angle -->
          <frameDir frame="%spacecraftFrameName%"
                    coord="%ofsetCoordType%"
                    units="%offsetFrameUnits%">%offsetCoords%</frameDir>
          <!-- Inertial reference direction for offset angle -->
          <baseFrameDir frame="%inertialFrameName%"
                        coord="%offsetBaseCoordType%"
```

```
                         units="%offsetBaseFrameUnits%">%offsetBaseCoord%</baseFrameDir>
           <projAngle localName="offsetAngle" />
         </offsetAngle>
         <angularRate localName="angularRate" />
       </attitude>
     </block>
   </definition>
 </metadata>
```

4.3.2.2 The variable content in the definitions template shall be substituted according to the rules in the following table. The values provided in the Tag column are those in the container: `/prm/body/segment/metadata/definition/`

Variable	Tag	Description	Allowed values	Example value
`%definitionName%`	`@name`	The identifier for the pointing elements definition; to be referenced in the generation of requests	-	
`%definitionVersion%`	`@version`	Version of the definition	By convention	`1.3`
`%inertialFrameName%`	`frame[1]/@name` `frame[2]/@baseframe`	Inertial reference frame name.	One of the inertial frames from ANNEX A.	`EME2000`
`%spaceraftFrameName%`	`frame/@name`	SC reference frame name	-	`SC`
`%OEM%`	`block/attitude/baseFrameDir/origin/orbitFile`	The URL to the orbit file containing the satellite trajectory (typically in OEM format)	-	`/home/SC/ephem.oem`
`%offsetBaseCoordType%`	`block/attitude/offsetAngle/baseFrameDir/@coord`	Type of coordinates defining the direction of the offset direction vector in inertial frame.	`cartesian` `spherical`	`cartesian`
`%offsetBaseFrameUnits%`	`block/attitude/offsetAngle/baseFrameDir/@units`	Units of the offset direction vector in inertial reference frame	For `%offsetBaseCoordType%=spherical:` `units="deg"` or `units="rad"` For `%offsetBaseCoordType%=cartesian` this variable must be an empty string.	`deg`

Variable	Tag	Description	Allowed values	Example value
%offsetBaseCoords%	block/attitude/offsetAngle/baseFrameDir	The value of the direction vector coordinates to be used as reference for the computation of the offset angle in inertial frame.	Any conforming to the direction type in Table 1	0. 0. 1.
%offsetCoordType%	block/attitude/offsetAngle/frameDir/@coord	Type of coordinates defining the direction of the offset direction vector in SC frame.	cartesian spherical	cartesian
%offsetFrameUnits%	block/attitude/offsetAngle/frameDir/@units	Units of the offset direction vector in SC reference frame	For %offsetCoordType%= spherical: units="deg" or units="rad" For %offsetCoordType%= cartesian this variable must be an empty string.	deg
%offsetCoords%	block/attitude/offsetAngle/frameDir	The value of the direction vector coordinates in SC frame to compute the offset angle with respect to the base offset coordinates	Any conforming to the direction type in Table 1	0. 1. 0.

4.3.3 REQUEST BODY TEMPLATE

4.3.3.1 The following template shall be used to build Sun pointing request blocks inside the PRM body. The variable content is shown between % symbols.

```
<data>
  <timeline frame="%spacecraftFrameName%">
    <block ref="sunPointing">
      <!-- Pointing request start time -->
      <blockStart>%blockStartEpoch%</blockStart>
      <!-- Pointing request end time -->
      <blockEnd>%blockEndEpoch%</blockEnd>
      <!-- SC axis to be pointed to Target -->
      <boresight frame="%spacecraftFrameName%"
                 coord="%spacecraftCoordType%"
                 units="%spacecraftFrameUnits%">%spacecraftAxisCoords%</boresight>
      <offsetAngle units="%offsetAngleUnits%">%offsetAngle%</offsetAngle>
      <angularRate units="%angularRateUnits%">%angularRate%</angularRate>
    </block>
  </timeline>
</data>
```

4.3.3.2 The variable content in the pointing request block template shall be substituted according to the rules in the following table. The values provided in the Tag column are those in the container: `/prm/body/segment/data/timeline/block/`

Variable	**Tag**	**Description**	**Allowed values**	**Example value**
`%blockStartEpoch%`	`blockStart`	Start epoch of the pointing request	Epoch according to 3.3.2.1	`2009-09-25T19:00:00.`
`%blockEndEpoch%`	`blockEnd`	End epoch of the pointing request	Epoch according to 3.3.2.1	`2009-09-25T20:00:00.`
`%spacecraftFrameName%`	`../@frame` `boresight/@frame`	SC reference frame name	-	`SC`
`%spacecraftCoordType%`	`boresight/@coord`	Coordinate type of the given pointed axis	cartesian spherical	`cartesian`
`%spacecraftFrameUnits%`	`boresight/@units`	Units of the direction vector in SC reference frame	For `%spacecraftCoordType%=spherical:` `units="deg"` or `units="rad"` For `% spacecraftCoordType %=cartesian` this variable must be an empty string.	`deg`
`%spacecraftAxisCoords%`	`Boresight`	Coordinates of the direction vector in the SC reference frame	Any conforming to the direction type in Table 1	`0.052336 0. 0.99863`
`%angularRateUnits%`	`angularRate/@units`	Units for the angular rate	`deg/s` `rad/s` `RPM`	`deg/s`
`%angularRate%`	`angularRate`	Angular rate value according to the real value representation in 3.3.2.6	-	`10.`
`%offsetAngleUnits%`	`offsetAngle/@units`	Units for the offset angle	`deg` `rad`	`deg`
`%offsetAngle%`	`offsetAngle`	Angle value according to the real value representation in 3.3.2.6	-	`10.`

4.4 TRACK WITH INERTIAL DIRECTION YAW STEERING

4.4.1 The track with inertial direction yaw steering shall be used to define a SC pointing request that fulfills the following conditions:

- a SC axis is pointed to a center of a solar system object.
- the remaining degree of freedom in the SC attitude is determined by a phase angle from a reference inertial direction to another SC axis.
- the SC axis and reference inertial direction used to define the phase shall not be parallel to the SC pointed axis and target direction respectively.
- the phase angle is the angle in the plane perpendicular to the target direction from the projection of the reference inertial direction to the projection of the SC axis, a positive angle meaning a positive rotation around the target direction. The resulting SC attitude is defined in ANNEX E.

4.4.2 DEFINITION FILE TEMPLATE

4.4.2.1 The following template shall be used to build the definitions for a PRM containing track with inertial direction yaw steering requests. The variable content is shown by variable names between % symbols.

```
<metadata>
  <TIME_SYSTEM>UTC</TIME_SYSTEM>
  <definition name="%definitionName%" version="%definitionVersion%" />
    <frame baseFrame="none" name="%inertialFrameName%" />
    <frame baseFrame="%inertialFrameName%" name="%spacecraftFrameName%" />
    <orbit name="%spacecraftName%">
      <!-- OEM containing the SC orbit  -->
      <orbitFile>%OEM%</orbitFile>
    </orbit>
    <orbit name="%targetBodyName%">
      <!-- The following two elements cannot appear together; one must be selected -->
      <!-- Either the object name for the reference target body ...   -->
      <ephObject>%targetBodyName%</ephObject>
      <!-- ... or the OEM containing the target object orbit -->
      <orbitFile>%targetOEM%</orbitFile>
    </orbit>
    <dirVector name="targetBody">
      <origin ref="%spacecraftName%"/>
      <target ref="%targetBodyName%"/>
    </dirVector>
    <block name="bodyTrackWithInertialYawSteering">
      <startEpoch localName="blockStart" />
      <endEpoch localName="blockEnd" />
      <attitude>
        <!-- Coordinates of default axis to be pointed -->
        <frameDir localName="boresight" />
        <baseFrameDir ref="targetBody" />
        <phaseAngle>
```

```
          <!-- SC reference direction for phase angle -->
          <frameDir frame="%spacecraftFrameName%"
                    coord="%phaseCoordType%"
                    units="%phaseFrameUnits%">%phaseCoords%</frameDir>
          <!-- Inertial reference direction for phase angle -->
          <baseFrameDir frame="%inertialFrameName%"
                        coord="%phaseBaseCoordType%"
                        units="%phaseBaseFrameUnits%">%phaseBaseCoord%</baseFrameDir>
          <projAngle localName="phaseAngle" />
        </phaseAngle>
      </attitude>
    </block>
  </definition>
</metadata>
```

4.4.2.2 The variable content in the definitions template shall be substituted according to the rules in the following table. The values provided in the Tag column are those in the container: `/prm/body/segment/metadata/definition/`

Variable	Tag	Description	Allowed values	Example value
`%definitionName%`	`@name`	The identifier for the pointing elements definition; to be referenced in the generation of requests	-	
`%definitionVersion%`	`@version`	Version of the definition	By convention	`1.3`
`%inertialFrameName%`	`frame[1]/@name` `frame[2]/@baseframe`	Inertial reference frame name.	One of the inertial frames from ANNEX A.	`EME2000`
`%spacecraftFrameName%`	`frame[2]/@name` `phaseAngle/frameDir/@frame`	SC reference frame name	-	`SC`
`%spacecraftName%`	`orbit[1]/@name` `dirVector[1]/origin/@name`	SC name	-	`MEX`
`%OEM%`	`orbit[1]/orbitFile`	The URL to the orbit file containing the satellite trajectory (typically in OEM format)		
`%targetBodyName%`	`orbit[2]/@name` `dirVector/target/@ref` `orbit[2]/ephObject`	The name of the body to be used as target for the pointing	Value given in [9]	`Mars`
`%targetOEM%`	`Orbit[2]/orbitFile`	The URL to the orbit file containing the trajectory of	Valid URL	

Variable	Tag	Description	Allowed values	Example value
		the target object (typically in OEM format)		
%phaseCoordType%	block/attitude/phaseAngle/frameDir/@coord	Type of coordinates defining the direction of the phase direction vector in SC frame.	cartesian spherical	cartesian
%phaseFrameUnits%	block/attitude/phaseAngle/frameDir/@units	Units of the phase direction vector in SC reference frame	For %phaseBaseCoordType%=spherical: units=”deg” or units=”rad” For %phaseBaseCoordType%=cartesian this variable must be an empty string.	deg
%phaseCoords%	block/attitude/phaseAngle/frameDir	The value of the direction vector coordinates in SC frame to compute the phase angle with respect to the base phase coordinates	Any conforming to the direction type in Table 1	0. 1. 0.
%phaseBaseCoordType%	block/attitude/phaseAngle/baseFrameDir/@coord	Type of coordinates defining the direction of the phase direction vector in inertial frame.	cartesian spherical	cartesian
%phaseBaseFrameUnits%	block/attitude/phaseAngle/baseFrameDir/@units	Units of the phase direction vector in inertial reference frame	For %phaseBaseCoordType%=spherical: units=”deg” or units=”rad” For %phaseBaseCoordType%=cartesian this variable must be an empty string.	deg
%phaseBaseCoords%	block/attitude/phaseAngle/baseFrameDir	The value of the direction vector coordinates to be used as reference for the computation of the phase angle in inertial frame.	Any conforming to the direction type in Table 1	0. 0. 1.

4.4.3 REQUEST BODY TEMPLATE

4.4.3.1 The following template shall be used to build track with inertial direction yaw steering request blocks inside the PRM body. The variable content is shown between % symbols.

```
<data>
  <timeline frame="%spacecraftFrameName%">
    <block ref="bodyTrackWithInertialYawSteering">
      <!-- Pointing request start time -->
      <blockStart>%blockStartEpoch%</blockStart>
      <!-- Pointing request end time -->
      <blockEnd>%blockEndEpoch%</blockEnd>
      <!-- SC axis to be pointed to the target body -->
      <boresight frame="%spacecraftFrameName%"
                 coord="%spacecraftCoordType%"
                 units="%spacecraftCoordUnits%">%spacecraftAxisCoords%</boresight>
      <targetBody ref="%tagetBodyName%" />
      <!-- Roll angle, see convention in Annex E  -->
      <phaseAngle units="%phaseAngleUnits%">%phaseAngle%</phaseAngle>
    </block>
  </timeline>
</data>
```

4.4.3.2 The variable content in the pointing request block template shall be substituted according to the rules in the following table. The values provided in the Tag column are those in the container: `/prm/body/segment/data/timeline/block/`

Variable	Tag	Description	Allowed values	Example value
`%spacecraftFrameName%`	`../@frame` `boresight/@frame`	SC reference frame name	-	`SC`
`%blockStartEpoch%`	`blockStart`	Start epoch of the pointing request	Epoch according to 3.3.2.1	`2009-09-25T19:00:00.`
`%blockEndEpoch%`	`blockEnd`	End epoch of the pointing request	Epoch according to 3.3.2.1	`2009-09-25T20:00:00.`
`%spacecraftCoordType%`	`boresight/@coord`	Coordinate type of the given pointed axis	`cartesian` `spherical`	`cartesian`
`%spacecraftCoordUnits%`	`boresight/@units`	Units of the SC axis to be kept aligned with relative phase to an inertial direction.	For `%phaseBaseCoordType%=spherical`: `units="deg"` or `units="rad"` For `%phaseBaseCoordType%=cartesian` this variable must be an empty string.	`deg`

Variable	Tag	Description	Allowed values	Example value
`%spacecraftAxisCoords%`	`boresight`	Coordinates of the direction vector in the SC reference frame	Any conforming to the direction type in Table 1	`0.052336 0. 0.99863`
`%targetBodyName%`	`targetBody/@ref`	The name of the target body to be pointed		`Mars`
`%phaseAngleUnits%`	`phaseAngle/@units`	Units for the phase angle	`deg` `rad`	`deg`
`%phaseAngle%`	`phaseAngle`	Angle value according to the real value representation in 3.3.2.6	-	`10.`

4.5 TRACK WITH POWER OPTIMIZED YAW STEERING

4.5.1 The track with power optimized yaw steering shall be used to define a SC pointing request that fulfills the following conditions:

- a SC axis is pointed to a center of a solar system object.
- a second SC axis is pointed in a direction perpendicular to the Sun direction such that this axis, the pointing direction and Sun direction are right handed.
- The two SC axes shall be perpendicular to each other.
- The Sun and direction shall not be parallel to the pointed axis for any instant of time of the pointing request.

4.5.2 DEFINITION FILE TEMPLATE

4.5.2.1 The following template shall be used to build the definitions for a PRM containing track with power optimised yaw steering requests. The variable content is shown by variable names between % symbols.

```
<metadata>
  <TIME_SYSTEM>UTC</TIME_SYSTEM>
  <definition name="%definitionName%" version="%definitionVersion%" />
    <frame baseFrame="none" name="%inertialFrameName%" />
    <frame baseFrame="%inertialFrameName%" name="%spacecraftFrameName%" />
    <orbit name="%spacecraftName%">
      <!-- OEM containing the SC orbit  -->
      <orbitFile>%OEM%</orbitFile>
    </orbit>
    <orbit name="%targetBodyName%">
      <!-- The following two elements cannot appear together; one must be selected -->
      <!-- Either the object name for the reference target body ... -->
      <ephObject>%targetBodyName%</ephObject>
      <!-- ... or the OEM containing the target object orbit -->
```

```
      <orbitFile>%targetOEM%</orbitFile>
    </orbit>
    <orbit name="Sun">
      <ephObject>SUN</ephObject>
    </orbit>
    <dirVector name="targetBody">
      <origin ref="%spacecraftName%"/>
      <target ref="%targetBodyName%"/>
    </dirVector>
    <dirVector name="Sun">
      <origin ref="%spacecraftName%"/>
      <target ref="Sun"/>
    </dirVector>
    <phaseAngle name="perpendicularToSun">
      <!-- Coordinates of SC axis to be kept perpendicular to Sun -->
      <!-- See signs convention on Annex E -->
      <frameDir frame="%spacecraftFrameName%"
                coord="%spacecraftCoordType%"

units="%spacecraftCoordUnits%">%spacecraftAxisPerpendicularToSun%</frameDir>
      <baseFrameDir ref="Sun" />
      <angle units="deg"> 90. </angle>
    </phaseAngle>
    <block name="bodyTrackWithPowerOptimisedYawSteering">
      <startEpoch localName="blockStart" />
      <endEpoch localName="blockEnd" />
      <attitude>
        <!-- Coordinates of default axis to be pointed -->
        <frameDir localName="boresight" />
        <baseFrameDir ref="targetBody" />
        <phaseAngle ref="perpendicularToSun" />
      </attitude>
    </block>
  </definition>
</metadata>
```

4.5.2.2 The variable content in the definitions template shall be substituted according to the rules in the following table. The values provided in the Tag column are those in the container: `/prm/body/segment/metadata/definition/`

Variable	**Tag**	**Description**	**Allowed values**	**Example value**
`%definitionName%`	`@name`	The identifier for the pointing elements definition; to be referenced in the generation of requests	-	
`%definitionVersion%`	`@version`	Version of the definition	By convention	`1.3`
`%inertialFrameName%`	`frame[1]/@name` `frame[2]/@baseframe`	Inertial reference frame name.	One of the inertial frames from ANNEX A.	`EME2000`

Variable	Tag	Description	Allowed values	Example value
%spacecraftFrameName%	frame[2]/@name phaseAngle/frameDir/@frame	SC reference frame name	-	SC
%spacecraftName%	orbit[1]/@name dirVector[1]/origin/@ref	SC name	-	MEX
%OEM%	orbit[1]/orbitFile	The URL to the orbit file containing the satellite trajectory (typically in OEM format)		
%targetBodyName%	orbit[2]/@name dirVector[1]/target/@ref orbit[2]/ephObject	The name of the body to be used as target for the pointing	Value given in [9]	Mars
%targetOEM%	Orbit[2]/orbitFile	The URL to the orbit file containing the trajectory of the target object (typically in OEM format)		
%spacecraftCoordType%	phaseAngle/frameDir/@coord	Coordinate type of the SC axis to be kept perpendicular to the Sun direction	cartesian spherical	cartesian
%spacecraftCoordUnits%	phaseAngle/frameDir/@units	Units of the SC axis to be kept perpendicular to the Sun direction	For %phaseBaseCoordType%=spherical: units="deg" or units="rad" For %phaseBaseCoordType%=cartesian this variable must be an empty string.	deg
%spacecraftAxisPerpendicularToSun%	phaseAngle/frameDir	Coordinates of the SC axis to be kept perpendicular to the Sun direction	-	0. 0. 1.

4.5.3 REQUEST BODY TEMPLATE

4.5.3.1 The following template shall be used to build track with power optimized yaw steering request blocks inside the PRM body. The variable content is shown between % symbols.

```
<data>
  <timeline frame="%spacecraftFrameName%">
    <block ref="bodyTrackWithPowerOptimised">
      <!-- Pointing request start time -->
      <blockStart>%blockStartEpoch%</blockStart>
      <!-- Pointing request end time -->
      <blockEnd>%blockEndEpoch%</blockEnd>
      <!-- SC axis to be pointed to the target body -->
      <boresight frame="%spacecraftFrameName%"
                 coord="%spacecraftCoordType%"
                 units="%spacecraftCoordUnits%">%spacecraftAxisCoords%</boresight>
    </block>
  </timeline>
</data>
```

4.5.3.2 The variable content in the pointing request block template shall be substituted according to the rules in the following table. The values provided in the Tag column are those in the container: `/prm/body/segment/data/timeline/block/`

Variable	Tag	Description	Allowed values	Example value
`%spacecraftFrameName%`	`../@frame` `boresight/@frame`	SC reference frame name	-	`SC`
`%blockStartEpoch%`	`blockStart`	Start epoch of the pointing request	Epoch according to 3.3.2.1	`2009-09-25T19:00:00.`
`%blockEndEpoch%`	`blockEnd`	End epoch of the pointing request	Epoch according to 3.3.2.1	`2009-09-25T20:00:00.`
`%spacecraftFrameName%`	`boresight/@frame`	SC reference frame name	-	`SC`
`%spacecraftCoordType%`	`boresight/@coord`	Coordinate type of the given pointed axis	cartesian spherical	`cartesian`
`%spacecraftCoordUnits%`	`phaseAngle/frameDir /@units`	Units of the SC axis to be kept perpendicular to the Sun direction	For `%spacecraftCoordType%`= `spherical`: `units="deg"` or `units="rad"` For `%spacecraftCoordType%`= `cartesian` this variable must be an empty string.	`deg`
`%spacecraftAxisCoords%`	`boresight`	Coordinates of the direction vector in the SC reference frame	Any conforming to the direction type in Table 1	`0.052336 0. 0.99863`

4.6 NADIR WITH POWER OPTIMIZED YAW STEERING

4.6.1 The nadir with power optimized yaw steering shall be used to define a SC pointing request that fulfills the following conditions:

- a SC axis is pointed such that the line along this axis intersects the surface of an object in nadir direction.
- a second SC axis is pointed in a direction perpendicular to the Sun direction such that this axis, the pointing direction and Sun direction form a right handed coordinate system.
- The two SC axes shall be perpendicular to each other.
- The Sun and Nadir direction shall not be parallel for any instant of time of the pointing request.

4.6.2 DEFINITION FILE TEMPLATE

4.6.2.1 The following template shall be used to build the definitions for a PRM containing nadir pointing with power optimized yaw steering requests. The variable content is shown by variable names between % symbols.

```
<metadata>
  <TIME_SYSTEM>UTC</TIME_SYSTEM>
  <definition name="%definitionName%" version="%definitionVersion%" />
    <frame baseFrame="none" name="%inertialFrameName%" />
    <frame baseFrame="%inertialFrameName%" name="%spacecraftFrameName%" />
    <orbit name="%spacecraftName%">
      <!-- OEM containing the SC orbit  -->
      <orbitFile>%OEM%</orbitFile>
    </orbit>
    <orbit name="Sun">
      <ephObject>SUN</ephObject>
    </orbit>
    <frame baseFrame="%inertialFrameName%" name="%targetBodyName%">
      <attitude>
        <!-- Planet reference frame -->
        <rotation from="%inertialFrameName%" to="%planetInertialFrame%" />
      </attitude>
    </frame>
    <orbit name="%targetBodyName%">
      <!-- Object name for the planet  -->
      <ephObject>%targetBodyName%</ephObject>
    </orbit>
    <surface name="nadirReferenceSurface" frame="%targetBodyName%">
      <origin ref="%targetBodyName%" />
      <!-- Planet reference ellipsoid -->
      <a units="%ellipsoidAxisUnits%">%ellipsoidSemiMajorAxis%</a>
      <b units="%ellipsoidAxisUnits%">%ellipsoidSemiMinorAxis%</b>
    </surface>
    <dirVector name="Sun">
      <origin ref="%spacecraftName%"/>
      <target ref="Sun"/>
```

```
    </dirVector>
    <dirVector name="nadir">
      <origin ref="%spacecraftName%" />
      <target>
        <surfaceVector operator="normal">
          <surface ref="nadirReferenceSurface" />
          <origin ref="%spacecraftName%" />
        </surfaceVector>
      </target>
    </dirVector>
    <phaseAngle name="perpendicularToSun">
      <!-- Coordinates of SC axis to be kept perpendicular to Sun -->
      <!-- See signs convention on Annex E -->
      <frameDir frame="%spacecraftFrameName%"
                coord="%spacecraftCoordType%"

units="%spacecraftCoordUnits%">%spacecraftAxisPerpendicularToSun%</frameDir>
      <baseFrameDir ref="Sun"/>
      <angle units="deg"> 90. </angle>
    </phaseAngle>
    <block name="nadir">
      <startEpoch localName="blockStart" />
      <endEpoch localName="blockEnd" />
      <attitude>
        <!-- Coordinates of default axis to be pointed -->
        <frameDir localName="boresight" />
        <baseFrameDir ref="nadir" />
        <phaseAngle ref="perpendicularToSun" />
      </attitude>
    </block>
  </definition>
</metadata>
```

4.6.2.2 The variable content in the definitions template shall be substituted according to the rules in the following table. The values provided in the Tag column are those in the container: `/prm/body/segment/metadata/definition/`

Variable	**Tag**	**Description**	**Allowed values**	**Example value**
`%definitionName%`	`@name`	The identifier for the pointing elements definition; to be referenced in the generation of requests	-	
`%definitionVersion%`	`@version`	Version of the definition	By convention	`1.3`
`%inertialFrameName%`	`frame[1]/@name` `frame[2]/@baseframe` `frame[3]/@baseframe` `frame[3]/attitude/rotation/from`	Inertial reference frame name.	One of the inertial frames from ANNEX A.	`EME2000`

Variable	Tag	Description	Allowed values	Example value
`%spacecraftFrameName%`	`frame[2]/@name` `phaseAngle/frameDir/@frame`	SC reference frame name	-	`SC`
`%spacecraftName%`	`orbit[1]/@name` `dirVector[1]/origin/@ref` `dirVector[1]/target/surfaceVector/origin/@ref` `dirVector[2]/origin/@ref` `dirVector[2]/target/surfaceVector/origin/@ref`	SC name	-	`MEX`
`%OEM%`	`orbit[1]/orbitFile`	The URL to the orbit file containing the satellite trajectory (typically in OEM format)		
`%targetBodyName%`	`frame[3]/@name` `orbit[3]/@name` `surface/origin/@ref` `stateVector[3]/ephObject`	The name of the body to be used as target for the pointing	Value given in [9]	`Mars`
`%planetInertialFrame%`	`frame[3]/attitude/rotation/@to`	Reference frame in the target body		`IAUMars`
`%ellipsoidAxisUnits%`	`surface/origin/a/@units` `surface/origin/b/@units`	Units for the dimension of the ellipsoid of the target body used to define the nadir pointing	km	`km`
`%ellipsoidSemiMajorAxis%`	`surface/origin/a`	Size of the semimajor axis of the ellipsoid of the target body	-	`6376.136`
`%ellipsoidSemiMinorAxis%`	`surface/origin/b`	Size of the semiminor axis of the ellipsoid of the target body	-	`6256.345`
`%spacecraftCoordType%`	`phaseAngle/frameDir/@coord`	Coordinate type of the SC axis to be kept perpendicular to the Sun direction	cartesian spherical	`cartesian`

Variable	**Tag**	**Description**	**Allowed values**	**Example value**
`%spacecraftCoordUnits%`	`phaseAngle/frameDir/@units`	Units of the SC axis to be kept perpendicular to the Sun direction	For %spacecraftCoordType%= `spherical`: `units="deg"` or `units="rad"` For %spacecraftCoordType%= `cartesian` this variable must be an empty string.	`deg`
`%spacecraftAxisPerpendicularToSun%`	`phaseAngle/frameDir`	Coordinates of the SC axis to be kept perpendicular to the Sun direction	-	`0. 0. 1.`

4.6.2.3 The direction vector type variables shall be given following the coordinates representation for direction vector type from section 3.3.2.9.

4.6.3 REQUEST BODY TEMPLATE

The variable content is shown between % symbols.

```
<data>
  <timeline frame="%spacecraftFrameName%">
    <block ref="nadir">
      <!-- Pointing request start time -->
      <blockStart>%blockStartEpoch%</blockStart>
      <!-- Pointing request end time -->
      <blockEnd>%blockEndEpoch%</blockEnd>
      <!-- SC axis to be pointed to Nadir -->
      <boresight frame="%spacecraftFrameName%"
                 coord="%spacecraftCoordType%"
                 units="%spacecraftCoordUnits%">%spacecraftAxisCoords%</boresight>
    </block>
  </timeline>
</data>
```

4.6.3.1 The variable content in the pointing request block template shall be substituted according to the rules in the following table. The values provided in the Tag column are those in the container: `/prm/body/segment/data/timeline/block/`

Variable	**Tag**	**Description**	**Allowed values**	**Example value**
`%spacecraftFrameName%`	`../@frame` `boresight/@frame`	SC reference frame name	-	`SC`
`%blockStartEpoch%`	`blockStart`	Start epoch of the pointing request	Epoch according to 3.3.2.1	`2009-09-25T19:00:00.`

Variable	Tag	Description	Allowed values	Example value
`%blockEndEpoch%`	`blockEnd`	End epoch of the pointing request	Epoch according to 3.3.2.1	`2009-09-25T20:00:00.`
`%spacecraftCoordType%`	`boresight/@coord`	Coordinate type of the given pointed axis	cartesian spherical	`cartesian`
`%spacecraftCoordUnits%`	`phaseAngle/frameDir/@units`	Units of the SC axis to be kept perpendicular to the Sun direction	For `% spacecraftCoordType %=spherical:` `units="deg"` or `units="rad"` For `% spacecraftCoordType %=cartesian` this variable must be an empty string.	`deg`
`%spacecraftAxisCoords%`	`boresight`	Coordinates of the direction vector in the SC reference frame	Any conforming to the direction type in Table 1	`0.052336 0. 0.99863`

4.6.3.2 The values for the base reference frame and SC reference frame names shall match the definitions.

4.6.3.3 The direction vector type variables (boresight and target direction) shall be given by its coordinates following the coordinates representation for direction vector type from section 3.3.2.9.

4.7 NADIR WITH GROUND TRACK ALIGNED YAW STEERING

4.7.1 The nadir with ground track aligned yaw steering templates in this section shall be used to define a SC pointing request that fulfills the following conditions:

- a SC axis is pointed such that the line along this axis intersects the surface of an object in nadir direction (e.g. relative to the reference surface provided for the computation, like the reference ellipsoid in the case of the Earth).
- a second SC axis is pointed perpendicular to the plane defined by nadir direction and the tangent to the ground track. The ground track is defined by the set of intersection points of the line along the SC pointed axis with the surface. The tangent to the ground track is defined in the surface fixed frame. The second SC axis, the nadir direction and the tangent in direction of increasing time shall shall form a right handed coordinate system.
- The two SC axes shall be perpendicular to each other.

- The ground track tangent in the surface fixed frame shall exist for any instant of time of the pointing request.

4.7.2 DEFINITION FILE TEMPLATE

4.7.2.1 The following template shall be used to build the definitions for a PRM containing nadir pointing with ground-track aligned yaw steering requests. The variable content is shown by variable names between % symbols.

```
<metadata>
  <TIME_SYSTEM>UTC</TIME_SYSTEM>
  <definition name="%definitionName%" version="%definitionVersion%" />
    <frame baseFrame="none" name="%inertialFrameName%" />
    <frame baseFrame="%inertialFrameName%" name="%spacecraftFrameName%" />
    <orbit name="%spacecraftName%">
      <!-- OEM containing the SC orbit  -->
      <orbitFile>%OEM%</orbitFile>
    </orbit>
    <frame baseFrame="%inertialFrameName%" name="%targetBodyName%">
      <attitude>
        <!-- Planet reference frame -->
        <rotation from="%inertialFrameName%" to="%planetInertialFrame%" />
      </attitude>
    </frame>
    <orbit name="%targetBodyName%">
      <!-- Object name for the planet  -->
      <ephObject>%targetBodyName%</ephObject>
    </orbit>
    <surface name="nadirReferenceSurface" frame="%targetBodyName%">
      <origin ref="%targetBodyName%" />
      <!-- Planet reference ellipsoid  -->
      <a units="%ellipsoidAxisUnits%">%ellipsoidSemiMajorAxis%</a>
      <b units="%ellipsoidAxisUnits%">%ellipsoidSemiMinorAxis%</b>
    </surface>
    <dirVector name="nadir">
      <origin ref="%spacecraftName%" />
      <target>
        <surfaceVector name="groundTrack" operator="normal">
          <surface ref="nadirReferenceSurface" />
          <origin ref="%spacecraftName%" />
        </surfaceVector>
      </target>
    </dirVector>
    <dirVector name="tangent" operator="tangent">
      <surfaceVector ref="groundTrack" />
    </dirVector>
    <phaseAngle name="perpendicularToGroundTrack">
      <!-- Coordinates of SC axis to be kept perpendicular to the ground track -->
      <!-- See signs convention on Annex E -->
      <frameDir frame="%spacecraftFrameName%"
                coord="%spacecraftCoordType%"

units="%spacecraftCoordUnits%">%spacecraftAxisPerpendicularToGroundTrack%</frameDir>
      <baseFrameDir ref="tangent"/>
      <angle units="deg"> 90. </angle>
```

```
    </phaseAngle>
    <block name="nadir">
      <startEpoch localName="blockStart" />
      <endEpoch localName="blockEnd" />
      <attitude>
        <!-- Coordinates of default axis to be pointed -->
        <frameDir localName="boresight" />
        <baseFrameDir ref="nadir" />
        <phaseAngle ref="perpendicularToGroundTrack" />
      </attitude>
    </block>
  </definition>
</metadata>
```

4.7.2.2 The variable content in the definitions template shall be substituted according to the rules in the following table. The values provided in the Tag column are those in the container: `/prm/body/segment/metadata/definition/`

Variable	Tag	Description	Allowed values	Example value
`%definitionName%`	`@name`	The identifier for the pointing elements definition; to be referenced in the generation of requests	-	
`%definitionVersion%`	`@version`	Version of the definition	By convention	`1.3`
`%inertialFrameName%`	`frame[1]/@name` `frame[2]/@baseframe` `frame[3]/@baseframe` `frame[3]/attitude/rotation/from`	Inertial reference frame name.	One of the inertial frames from ANNEX A.	`EME2000`
`%spacecraftFrameName%`	`frame[2]/@name` `phaseAngle/frameDir/@frame`	SC reference frame name	-	`SC`
`%spacecraftName%`	`orbit[1]/@name` `dirVector[1]/origin/@ref` `dirVector[1]/target/surfaceVector/origin/@ref`	SC name	-	`MEX`
`%OEM%`	`orbit[1]/orbitFile`	The URL to the orbit file containing the satellite trajectory (typically in OEM format)		

Variable	Tag	Description	Allowed values	Example value
%targetBodyName%	frame[3]/@name orbit[3]/@name surface/@frame surface/origin/@ref orbit[2]/ephObject	The name of the body to be used as target for the pointing	Value given in [9]	Mars
%planetInertialFrame%	frame[3]/attitude/rotation/@to	Reference frame in the target body		IAUMars
%ellipsoidAxisUnits%	surface/origin/a/@units surface/origin/b/@units	Units for the dimension of the ellipsoid of the target body used to define the nadir pointing	km	km
%ellipsoidSemiMajorAxis%	surface/origin/a	Size of the semimajor axis of the ellipsoid of the target body	-	6376.136
%ellipsoidSemiMinorAxis%	surface/origin/b	Size of the semiminor axis of the ellipsoid of the target body	-	6256.345
%spacecraftCoordType%	phaseAngle/frameDir/@coord	Coordinate type of the SC axis to be kept perpendicular to the ground track	cartesian spherical	cartesian
%spacecraftCoordUnits%	phaseAngle/frameDir/@units	Units of the SC axis to be kept perpendicular to the ground track	For %spacecraftCoordType%= spherical: units="deg" or units="rad" For %spacecraftCoordType%= cartesian this variable must be an empty string.	deg
%spacecraftAxisPerpendicularToGroundTrack%	phaseAngle/frameDir	Coordinates of the SC axis to be kept perpendicular to the the ground track	-	0. 0. 1.

4.7.2.3 The direction vector type variables shall be given by its coordinates following the coordinates representation for direction vector type from section 3.3.2.9.

4.7.3 REQUEST BODY TEMPLATE

4.7.3.1 The following template shall be used to build nadir pointing with ground-track aligned yaw steering request blocks inside the PRM body. The variable content is shown between % symbols.

```
<data>
  <timeline frame="%spacecraftFrameName%">
    <block ref="nadir">
      <!-- Pointing request start time -->
      <blockStart>%blockStartEpoch%</blockStart>
      <!-- Pointing request end time -->
      <blockEnd>%blockEndEpoch%</blockEnd>
      <!-- SC axis to be pointed to Nadir -->
      <boresight frame="%spacecraftFrameName%"
                 coord="%spacecraftCoordType%"
                 units="%spacecraftCoordUnits%">%spacecraftAxisCoords%</boresight>
    </block>
  </timeline>
</data>
```

4.7.3.2 The variable content in the pointing request block template shall be substituted according to the rules in the following table. The values provided in the Tag column are those in the container: `/prm/body/segment/data/timeline/block/`

Variable	Tag	Description	Allowed values	Example value
`%spacecraftFrameName%`	`../@frame` `boresight/@frame`	SC reference frame name	-	`SC`
`%blockStartEpoch%`	`blockStart`	Start epoch of the pointing request	Epoch according to 3.3.2.1	`2009-09-25T19:00:00.`
`%blockEndEpoch%`	`blockEnd`	End epoch of the pointing request	Epoch according to 3.3.2.1	`2009-09-25T20:00:00.`
`%spacecraftCoordType%`	`boresight/@coord`	Coordinate type of the given pointed axis	cartesian spherical	`cartesian`
`%spacecraftCoordUnits%`	`boresight/@units`	Units of the given pointed axis.	For `%spacecraftCoordType%=spherical`: `units="deg"` or `units="rad"` For `%spacecraftCoordType%=cartesian` this variable must be an empty string.	`deg`

Variable	Tag	Description	Allowed values	Example value
`%spacecraftAxisCoords%`	`boresight`	Coordinates of the direction vector in the SC reference frame	Any conforming to the direction type in Table 1	`0.052336 0. 0.99863`

4.7.3.3 The values for the base reference frame and SC reference frame names shall match the definitions.

4.7.3.4 The direction vector type variables (boresight and target direction) shall be given by its coordinates following the coordinates representation for direction vector type from section 3.3.2.9.

4.8 NADIR WITH ORBITAL POLE ALIGNED YAW STEERING

4.8.1 The nadir with orbital pole aligned yaw steering shall be used to define a SC pointing request that fulfills the following conditions:

- a SC axis is pointed such that the line along this axis intersects the surface of an object in nadir direction (e.g. relative to the reference surface provided for the computation, like the reference ellipsoid in the case of the Earth).

- a second SC axis is aligned with the SC orbital pole with respect to the object projected in the plane perpendicular to the nadir direction.

- The two SC axes shall be perpendicular to each other.

4.8.2 DEFINITION FILE TEMPLATE

4.8.2.1 The following template shall be used to build the definitions for a PRM containing nadir pointing with orbital pole aligned yaw steering requests. The variable content is shown by variable names between % symbols.

```
<metadata>
  <TIME_SYSTEM>UTC</TIME_SYSTEM>
  <definition name="%definitionName%" version="%definitionVersion%" />
    <frame baseFrame="none" name="%inertialFrameName%" />
    <frame baseFrame="%inertialFrameName%" name="%spacecraftFrameName%" />
    <orbit name="%spacecraftName%">
      <!-- OEM containing the SC orbit  -->
      <orbitFile>%OEM%</orbitFile>
    </orbit>
    <frame baseFrame="%inertialFrameName%" name="%targetBodyName%">
      <attitude>
        <!-- Planet reference frame -->
        <rotation from="%inertialFrameName%" to="%planetInertialFrame%" />
      </attitude>
    </frame>
    <orbit name="%targetBodyName%">
      <!-- Object name for the planet  -->
      <ephObject>%targetBodyName%</ephObject>
```

```
    </orbit>
    <surface name="nadirReferenceSurface" frame="%targetBodyName%">
      <origin ref="%targetBodyName%" />
      <!-- Planet reference ellipsoid  -->
      <a units="%ellipsoidAxisUnits%">%ellipsoidSemiMajorAxis%</a>
      <b units="%ellipsoidAxisUnits%">%ellipsoidSemiMinorAxis%</b>
    </surface>
    <dirVector name="nadir">
      <origin ref="%spacecraftName%" />
      <target>
        <surfaceVector operator="normal">
          <surface ref="nadirReferenceSurface" />
          <origin ref="%spacecraftName%" />
        </surfaceVector>
      </target>
    </dirVector>
    <dirVector name="orbitalPole" operator="cross">
      <dirVector name="scToTargetBody">
        <origin ref="%spacecraftName%" />
        <target ref="%targetBodyName%" />
      </dirVector>
      <dirVector ref="scToTargetBody" operator="derivative " />
    </dirVector>
    <phaseAngle name="alignedWithOrbitalPole">
      <!-- Coordinates of SC axis to be kept perpendicular to the ground track -->
      <!-- See signs convention on Annex E -->
      <frameDir frame="%spacecraftFrameName%"
                coord="%spacecraftCoordType%"

units="%spacecraftCoordUnits%">%spacecraftAxisParallelToOrbitPole%</frameDir>
      <baseFrameDir ref="orbitalPole"/>
      <projAngle units="deg"> 0. </angle>
    </phaseAngle>
    <block name="nadir">
      <startEpoch localName="blockStart" />
      <endEpoch localName="blockEnd" />
      <attitude>
        <!-- Coordinates of default axis to be pointed -->
        <frameDir localName="boresight" />
        <baseFrameDir ref="nadir" />
            <phaseAngle ref="alignedWithOrbitalPole" />
      </attitude>
    </block>
  </definition>
</metadata>
```

4.8.2.2 The variable content in the definitions template shall be substituted according to the rules in the following table. The values provided in the Tag column are those in the container: `/prm/body/segment/metadata/definition/`

Variable	Tag	Description	Allowed values	Example value
%definitionName%	@name	The identifier for the pointing elements definition;	-	

Variable	Tag	Description	Allowed values	Example value
		to be referenced in the generation of requests		
%definitionVersion%	@version	Version of the definition	By convention	1.3
%inertialFrameName%	frame[1]/@name frame[2]/@baseframe frame[3]/@baseframe frame[3]/attitude/rotation/@from	Inertial reference frame name.	One of the inertial frames from ANNEX A.	EME2000
%spacecraftFrameName%	frame[2]/@name phaseAngle/frameDir/@frame	SC reference frame name	-	SC
%spacecraftName%	orbit[1]/@name dirVector[1]/origin/@ref dirVector[1]/target/surfaceVector/origin/@ref dirVector[2]/dirVector/origin/@ref	SC name	-	MEX
%OEM%	orbit[1]/orbitFile	The URL to the orbit file containing the satellite trajectory (typically in OEM format)		
%targetBodyName%	frame[3]/@name orbit[2]/@name surface/@frame surface/origin/@ref orbit[2]/ephObject	The name of the body to be used as target for the pointing	Value given in [9]	Mars
%planetInertialFrame%	frame[3]/attitude/rotation/@to	Reference frame in the target body		IAUMars
%ellipsoidAxisUnits%	surface/origin/a/@units surface/origin/b/@units	Units for the dimension of the ellipsoid of the target body used to define the nadir pointing	km	km

Variable	Tag	Description	Allowed values	Example value
%ellipsoidSemiMajorAxis%	surface/origin/a	Size of the semimajor axis of the ellipsoid of the target body	-	6376.136
%ellipsoidSemiMinorAxis%	surface/origin/b	Size of the semiminor axis of the ellipsoid of the target body	-	6256.345
%spacecraftCoordType%	phaseAngle/frameDir/@coord	Coordinate type of the SC axis to be kept parallel to the orbit pole	cartesian spherical	cartesian
%spacecraftCoordUnits%	phaseAngle/frameDir/@units	Units of the SC axis to be kept parallel to the orbit pole	For %spacecraftCoordType%= spherical: units="deg" or units="rad" For %spacecraftCoordType%= cartesian this variable must be an empty string.	deg
%spacecraftAxisParallelToOrbitPole%	phaseAngle/frameDir	Coordinates of the SC axis to be kept parallel to the orbit pole	-	0. 0. 1.

4.8.2.3 The direction vector type variables shall be given by its coordinates following the coordinates representation for direction vector type from section 3.3.2.9.

4.8.3 REQUEST BODY TEMPLATE

4.8.3.1 The following template shall be used to build nadir pointing with orbital pole aligned yaw steering request blocks inside the PRM body. The variable content is shown between % symbols.

```
<data>
  <timeline frame="%spacecraftFrameName%">
    <block ref="nadir">
      <!-- Pointing request start time -->
      <blockStart>%blockStartEpoch%</blockStart>
      <!-- Pointing request end time -->
      <blockEnd>%blockEndEpoch%</blockEnd>
      <!-- SC axis to be pointed to Nadir -->
      <boresight frame="%spacecraftFrameName%"
                 coord="%spacecraftCoordType%"
                 units="%spacecraftCoordUnits%">%spacecraftBoreCoords%</boresight>
    </block>
  </timeline>
```

```
</data>
```

4.8.3.2 The variable content in the pointing request block template shall be substituted according to the rules in the following table. The values provided in the Tag column are those in the container: `/prm/body/segment/data/timeline/block/`

Variable	Tag	Description	Allowed values	Example value
`%spacecraftFrameName%`	`../@frame` `boresight/@frame`	SC reference frame name	-	`SC`
`%blockStartEpoch%`	`blockStart`	Start epoch of the pointing request	Epoch according to 3.3.2.1	`2009-09-25T19:00:00.`
`%blockEndEpoch%`	`blockEnd`	End epoch of the pointing request	Epoch according to 3.3.2.1	`2009-09-25T20:00:00.`
`%spacecraftCoordType%`	`boresight/@coord`	Coordinate type of the given pointed axis	cartesian spherical	`cartesian`
`%spacecraftCoordUnits%`	`boresight/@units`	Units of the given pointed axis	For `% spacecraftCoordType %=spherical:` `units="deg"` or `units="rad"` For `% spacecraftCoordType %=cartesian` this variable must be an empty string.	`deg`
`%spacecraftBoreCoords%`	`boresight`	Coordinates of the direction vector in the SC reference frame for the main pointing axis	Any conforming to the direction type in Table 1	`0.052336 0. 0.99863`

4.9 LIMB POINTING WITH POWER OPTIMIZED YAW STEERING

4.9.1 The limb pointing with power optimized yaw steering template shall be used to define a SC pointing request that fulfills the following conditions:

- a SC axis is pointed towards a point that lies at a specified height along the local normal over a point on the limb of an object.
- The point on the limb is defined as the intersection point of the limb with a half-plane defined by the SC to object-center direction and a positive component towards a reference inertial direction.

- The reference inertial direction shall not be aligned with the SC to object-center direction
- a second SC axis is pointed in a direction perpendicular to the Sun direction such that this axis, the pointing direction and Sun direction are right handed.
- The two SC axes shall be perpendicular to each other.
- The Sun and target direction shall not be parallel for any instant of time of the pointing request.

4.9.2 DEFINITION FILE TEMPLATE

4.9.2.1 The following template shall be used to build the definitions for a PRM containing limb pointing with power optimized yaw steering requests. The variable content is shown by variable names between % symbols.

```
<metadata>
  <TIME_SYSTEM>UTC</TIME_SYSTEM>
  <definition name="%definitionName%" version="%definitionVersion%" />
    <frame baseFrame="none" name="%inertialFrameName%" />
    <frame baseFrame="%inertialFrameName%" name="%spacecraftFrameName%" />
    <orbit name="%spacecraftName%">
      <!-- OEM containing the SC orbit  -->
      <orbitFile>%OEM%</orbitFile>
    </orbit>
    <orbit name="%targetBodyName%">
      <!-- Object name for the reference target body  -->
      <ephObject>%targetBodyName%</ephObject>
    </orbit>
    <orbit name="Sun">
      <ephObject>SUN</ephObject>
    </orbit>
    <frame baseFrame="%inertialFrameName%" name="%targetBodyName%">
      <attitude>
        <!-- Planet reference frame -->
        <rotation from="%inertialFrameName%" to="%planetInertialFrame%" />
      </attitude>
    </frame>
    <surface name="limbReferenceSurface" frame="%targetBodyName%">
      <origin ref="%targetBodyName%" />
      <!-- Planet reference ellipsoid  -->
      <a units="%ellipsoidAxisUnits%">%ellipsoidSemiMajorAxis%</a>
      <b units="%ellipsoidAxisUnits%">%ellipsoidSemiMinorAxis%</b>
    </surface>
    <dirVector name="Sun">
      <origin ref="%spacecraftName%"/>
      <target ref="Sun"/>
    </dirVector>
    <phaseAngle name="perpendicularToSun">
      <!-- Coordinates of SC axis to be kept perpendicular to Sun -->
      <!-- See signs convention on Annex E -->
      <frameDir frame="%spacecraftFrameName%"
```

```
                 coord="%spacecraftCoordType%"

units="%spacecraftCoordUnits%">%spacecraftAxisPerpendicularToSun%</frameDir>
      <baseFrameDir ref="Sun"/>
      <angle units="deg"> 90. </angle>
    </phaseAngle>
    <block name="limbWithPowerOptimisedYawSteering">
      <startEpoch localName="blockStart" />
      <endEpoch localName="blockEnd" />
      <attitude>
        <!-- Coordinates of default axis to be pointed -->
        <frameDir localName="boresight" />
        <baseFrameDir ref="target" />
        <phaseAngle ref="perpendicularToSun" />
        <surfaceVector localName="target">
      </attitude>
    </block>
  </definition>
</metadata>
```

4.9.2.2 The variable content in the definitions template shall be substituted according to the rules in the following table. The values provided in the Tag column are those in the container: `/prm/body/segment/metadata/definition/`

Variable	Tag	Description	Allowed values	Example value
`%definitionName%`	`@name`	The identifier for the pointing elements definition; to be referenced in the generation of requests	-	
`%definitionVersion%`	`@version`	Version of the definition	By convention	`1.3`
`%inertialFrameName%`	`frame[1]/@name` `frame[2]/@baseframe` `frame[3]/@baseframe` `frame[2]/@baseframe/atttude/rotation/@from`	Inertial reference frame name.	One of the inertial frames from ANNEX A.	`EME2000`
`%spacecraftFrameName%`	`frame[2]/@name` `phaseAngle/frameDir/@frame`	SC reference frame name	-	`SC`
`%spacecraftName%`	`orbit[1]/@name` `dirVector/origin`	SC name	-	`MEX`
`%OEM%`	`orbit[1]/orbitFile`	The URL to the orbit file containing the satellite trajectory (typically in OEM format)		

Variable	Tag	Description	Allowed values	Example value
%targetBodyName%	orbit[2]/@name frame[3]/@name surface/@frame surface/origin/@ref orbit[2]/ephObject	The name of the body to be used as target for the pointing	Value given in [9]	Mars
%planetInertialFrame%	frame[3]/attitude/rotation/@to	Reference frame in the target body		IAUMars
%ellipsoidAxisUnits%	surface/origin/a/@units surface/origin/b/@units	Units for the dimension of the ellipsoid of the target body used to define the nadir pointing	km	km
%ellipsoidSemiMajorAxis%	surface/origin/a	Size of the semimajor axis of the ellipsoid of the target body	-	6376.136
%ellipsoidSemiMinorAxis%	surface/origin/b	Size of the semiminor axis of the ellipsoid of the target body	-	6256.345
%sufaceCoords%	surfaceVector/surfaceCoord	The coordinates of the surface limb point to use as reference for the target	-	52.3 65.4
%spacecraftCoordType%	phaseAngle/frameDir/@coord	Coordinate type of the SC axis to be kept perpendicular to the Sun direction	cartesian spherical	Cartesian
%spacecraftCoordUnits%	phaseAngle/frameDir/@units	Units of the SC axis to be kept perpendicular to the Sun direction	For %spacecraftCoordType%= spherical: units="deg" or units="rad" For %spacecraftCoordType%= cartesian this variable must be an empty string.	deg
%spacecraftAxisPerpendicularToSun%	phaseAngle/frameDir	Coordinates of the SC axis to be kept perpendicular to the Sun direction	-	0. 0. 1.

4.9.3 REQUEST BODY TEMPLATE

4.9.3.1 The following template shall be used to build limb pointing with power optimized yaw steering request blocks inside the PRM body. The variable content is shown between % symbols.

```
<data>
  <timeline frame="%spacecraftFrameName%">
    <block ref="limbWithPowerOptimisedYawSteering">
      <!-- Pointing request start time -->
      <blockStart>%blockStartEpoch%</blockStart>
      <!-- Pointing request end time -->
      <blockEnd>%blockEndEpoch%</blockEnd>
      <!-- SC axis to be pointed to the target body -->
      <boresight frame="%spacecraftFrameName%"
                 coord="%spacecraftCoordType%"
                 units="%spacecraftCoordUnits%">%spacecraftAxisCoords%</boresight>
      <target>
          <surface ref="limbReferenceSurface" />
          <dirVector coord="%limbCoordType%"
                     units="%limbCoordUnits%">%limbCoord%</dirVector>
          <height units ="%heightUnits%">%height%</height>
      </target>
    </block>
  </timeline>
</data>
```

4.9.3.2 The variable content in the pointing request block template shall be substituted according to the rules in the following table. The values provided in the Tag column are those in the container: `/prm/body/segment/data/timeline/block/`

Variable	Tag	Description	Allowed values	Example value
`%spacecraftFrameName%`	`../@frame` `boresight/@frame`	SC reference frame name	-	`SC`
`%blockStartEpoch%`	`blockStart`	Start epoch of the pointing request	Epoch according to 3.3.2.1	`2009-09-25T19:00:00.`
`%blockEndEpoch%`	`blockEnd`	End epoch of the pointing request	Epoch according to 3.3.2.1	`2009-09-25T20:00:00.`
`%spacecraftCoordType%`	`boresight/@coord`	Coordinate type of the given pointed axis	`cartesian` `spherical`	`cartesian`
`%spacecraftCoordUnits%`	`boresight/@units`	Units of the given pointed axis	For %spacecraftCoordType% `=spherical`: `units="deg"` or `units="rad"` For `%phaseBaseCoordType%=cartesian`	`deg`

Variable	Tag	Description	Allowed values	Example value
			this variable must be an empty string.	
%spacecraftAxisCoords%	Boresight	Coordinates of the direction vector in the SC reference frame	Any conforming to the direction type in Table 1	0.052336 0. 0.99863
%limbCoordType%	target/dirVector/@coord	Coordinate type of the limb coordinates to be used as reference for the pointing	cartesian spherical	cartesian
%limbCoordUnits%	target/dirVector/@units	Units of the limb coordinates to be used as reference for the pointing	For %limbCoordType%=spherical: units="deg" or units="rad" For %limbCoordType%=cartesian this variable must be an empty string.	deg
%limbCoord%	target/dirVector	Limb coordinates to be used as reference for the pointing	-	235.5 3.25
%heightUnits%	target/height/@units	The units of the height above the limb point to use for the calculation of the target	-	km
%height%	target/height	The height above the limb point to use for the calculation of the target	-	124.7

4.10 LIMB POINTING WITH INERTIAL DIRECTION YAW STEERING

4.10.1 The limb pointing with inertial direction yaw steering template shall be used to define a SC pointing request that fulfills the following conditions:

- a SC axis is pointed towards a point that lies at a specified height along the local normal over a point on the limb of an object.
- The point on the limb is defined as intersection point of the limb with a half-plane defined by the SC to object-center direction and a positive component towards a reference inertial direction.
- The reference inertial direction shall not be aligned with the SC to object-center direction

- the remaining degree of freedom in the SC attitude is determined by a phase angle from the reference inertial direction to another SC axis.
- the SC axis and reference inertial direction used to define the phase shall not be parallel to the SC pointed axis and target direction respectively.
- the phase angle is the angle in the plane perpendicular to the target direction from the projection of the reference inertial direction to the projection of the SC axis, a positive angle meaning a positive rotation around the target direction. The resulting SC attitude is defined in ANNEX E.

4.10.2 DEFINITION FILE TEMPLATE

4.10.2.1 The following template shall be used to build the definitions for a PRM containing limb pointing with inertial direction yaw steering requests. The variable content is shown by variable names between % symbols.

```
<metadata>
  <TIME_SYSTEM>UTC</TIME_SYSTEM>
  <definition name="%definitionName%" version="%definitionVersion%" />
    <frame baseFrame="none" name="%inertialFrameName%" />
    <frame baseFrame="%inertialFrameName%" name="%spacecraftFrameName%" />
    <orbit name="%spacecraftName%">
      <!-- OEM containing the SC orbit  -->
      <orbitFile>%OEM%</orbitFile>
    </orbit>
    <orbit name="%targetBodyName%">
      <!-- Object name for the reference target body  -->
      <ephObject>%targetBodyName%</ephObject>
    </orbit>
    <frame baseFrame="%inertialFrameName%" name="%targetBodyName%">
      <attitude>
        <!-- Planet reference frame -->
        <rotation from="%inertialFrameName%" to="%planetInertialFrame%" />
      </attitude>
    </frame>
    <surface name="limbReferenceSurface" frame="%targetBodyName%">
      <origin ref="%targetBodyName%" />
      <!-- Planet reference ellipsoid  -->
      <a units="%ellipsoidAxisUnits%">%ellipsoidSemiMajorAxis%</a>
      <b units="%ellipsoidAxisUnits%">%ellipsoidSemiMinorAxis%</b>
    </surface>
    <!-- Inertial reference direction for phase angle -->
    <phaseAngle>
      <!-- SC reference direction for phase angle -->
      <frameDir frame="%spacecraftFrameName%"
                coord="%phaseCoordType%"
                units="%phaseFrameUnits%">%phaseCoords%</frameDir>
      <!-- Inertial reference direction for phase angle -->
      <baseFrameDir frame="%inertialFrameName%"
                    coord="%phaseBaseCoordType%"
                    units="%phaseBaseFrameUnits%">%phaseBaseCoord%</baseFrameDir>
      <projAngle localName="phaseAngle" />
```

```
    </phaseAngle>
    <block name="limbWithPowerOptimisedYawSteering">
      <startEpoch localName="blockStart" />
      <endEpoch localName="blockEnd" />
      <attitude>
        <!-- Coordinates of default axis to be pointed -->
        <frameDir localName="boresight" />
        <baseFrameDir ref="target" />
        <surfaceVector localName="target">
      </attitude>
    </block>
  </definition>
</metadata>
```

4.10.2.2 The variable content in the definitions template shall be substituted according to the rules in the following table. The values provided in the Tag column are those in the container: `/prm/body/segment/metadata/definition/`

Variable	Tag	Description	Allowed values	Example value
`%definitionName%`	`@name`	The identifier for the pointing elements definition; to be referenced in the generation of requests	-	
`%definitionVersion%`	`@version`	Version of the definition	By convention	`1.3`
`%inertialFrameName%`	`frame[1]/@name` `frame[2]/@baseframe` `frame[3]/@baseframe` `frame[2]/@baseframe/atttude/rotation/@from`	Inertial reference frame name.	One of the inertial frames from ANNEX A.	`EME2000`
`%spacecraftFrameName%`	`frame[2]/@name` `phaseAngle/frameDir/@frame`	SC reference frame name	-	`SC`
`%spacecraftName%`	`orbit[1]/@name` `dirVector/origin`	SC name	-	`MEX`
`%OEM%`	`orbit[1]/orbitFile`	The URL to the orbit file containing the satellite trajectory (typically in OEM format)		
`%targetBodyName%`	`orbit[2]/@name` `frame[3]/@name` `surface/@frame`	The name of the body to be used as target for the pointing	Value given in [9]	`Mars`

Variable	Tag	Description	Allowed values	Example value
	surface/origin/@ref orbit[2]/ephObject			
%planetInertialFrame%	frame[3]/attitude/rotation/@to	Reference frame in the target body		IAUMars
%ellipsoidAxisUnits%	surface/origin/a/@units surface/origin/b/@units	Units for the dimension of the ellipsoid of the target body used to define the nadir pointing	km	km
%ellipsoidSemiMajorAxis%	surface/origin/a	Size of the semimajor axis of the ellipsoid of the target body	-	6376.136
%ellipsoidSemiMinorAxis%	surface/origin/b	Size of the semiminor axis of the ellipsoid of the target body	-	6256.345
%sufaceCoords%	surfaceVector/surfaceCoord	The coordinates of the surface limb point to use as reference for the target	-	52.3 65.4
%phaseCoordType%	block/attitude/phaseAngle/frameDir/@coord	Type of coordinates defining the direction of the phase direction vector in SC frame.	cartesian spherical	cartesian
%phaseFrameUnits%	block/attitude/phaseAngle/frameDir/@units	Units of the phase direction vector in SC reference frame	For %phaseBaseCoordType%=spherical: units=”deg” or units=”rad” For %phaseBaseCoordType%=cartesian this variable must be an empty string.	deg
%phaseCoords%	block/attitude/phaseAngle/frameDir	The value of the direction vector coordinates in SC frame to compute the phase angle with respect to the base phase coordinates	Any conforming to the direction type in Table 1	0. 1. 0.
%phaseBaseCoordType%	block/attitude/phaseAngle/baseFrameDir/@coord	Type of coordinates defining the direction of the phase direction vector in inertial frame.	cartesian spherical	cartesian

Variable	Tag	Description	Allowed values	Example value
`%phaseBaseFrameUnits%`	`block/attitude/phaseAngle/baseFrameDir/@units`	Units of the phase direction vector in inertial reference frame	For %phaseBaseCoordType%=spherical: units=”deg” or units=”rad” For %phaseBaseCoordType%=cartesian this variable must be an empty string.	`deg`
`%phaseBaseCoords%`	`block/attitude/phaseAngle/baseFrameDir`	The value of the direction vector coordinates to be used as reference for the computation of the phase angle in inertial frame.	Any conforming to the direction type in Table 1	`0. 0. 1.`

4.10.3 REQUEST BODY TEMPLATE

4.10.3.1 The following template shall be used to build limb pointing with inertial direction yaw steering request blocks inside the PRM body. The variable content is shown between % symbols.

```
<data>
  <timeline frame="%spacecraftFrameName%">
    <block ref="limbWithPowerOptimisedYawSteering">
      <!-- Pointing request start time -->
      <blockStart>%blockStartEpoch%</blockStart>
      <!-- Pointing request end time -->
      <blockEnd>%blockEndEpoch%</blockEnd>
      <!-- SC axis to be pointed to the target body -->
      <boresight frame="%spacecraftFrameName%"
                 coord="%spacecraftCoordType%"
                 units="%spacecraftCoordUnits%">%spacecraftAxisCoords%</boresight>
      <target>
          <surface ref="limbReferenceSurface" />
          <dirVector coord="%limbCoordType%"
                     units="%limbCoordUnits%">%limbCoord%</dirVector>
          <height units ="%heightUnits%">%height%</height>
      </target>
      <!-- Roll angle, see convention in Annex E  -->
      <phaseAngle units="%phaseAngleUnits%">%phaseAngle%</phaseAngle>
    </block>
  </timeline>
</data>
```

4.10.3.2 The variable content in the pointing request block template shall be substituted according to the rules in the following table. The values provided in the Tag column are those in the container: `/prm/body/segment/data/timeline/block/`

Variable	Tag	Description	Allowed values	Example value
%spacecraftFrameName%	../@frame boresight/@frame	SC reference frame name	-	SC
%blockStartEpoch%	blockStart	Start epoch of the pointing request	Epoch according to 3.3.2.1	2009-09-25T19:00:00.
%blockEndEpoch%	blockEnd	End epoch of the pointing request	Epoch according to 3.3.2.1	2009-09-25T20:00:00.
%spacecraftCoordType%	boresight/@coord	Coordinate type of the given pointed axis	cartesian spherical	cartesian
%spacecraftCoordUnits%	boresight/@units	Units of the given pointed axis	For %spacecraftCoordType%=spherical: units="deg" or units="rad" For %phaseBaseCoordType%=cartesian this variable must be an empty string.	deg
%spacecraftAxisCoords%	Boresight	Coordinates of the direction vector in the SC reference frame	Any conforming to the direction type in Table 1	0.052336 0. 0.99863
%limbCoordType%	target/dirVector/@coord	Coordinate type of the limb coordinates to be used as reference for the pointing	cartesian spherical	cartesian
%limbCoordUnits%	target/dirVector/@units	Units of the limb coordinates to be used as reference for the pointing	For %limbCoordType%=spherical: units="deg" or units="rad" For %limbCoordType%=cartesian this variable must be an empty string.	deg
%limbCoord%	target/dirVector	Limb coordinates to be used as reference for the pointing	-	235.5 3.25
%heightUnits%	target/height/@units	The units of the height above the limb point to use for the calculation of the target	-	km

Variable	Tag	Description	Allowed values	Example value
`%height%`	`target/height`	The height above the limb point to use for the calculation of the target	-	`124.7`
`%phaseAngleUnits%`	`phaseAngle/@units`	Units for the phase angle	deg rad	`deg`
`%phaseAngle%`	`phaseAngle`	Angle value according to the real value representation in 3.3.2.6	-	`10.`

4.11 VELOCITY POINTING WITH ORBITAL POLE YAW STEERING

4.11.1 The velocity pointing with orbital pole yaw steering template shall be used to define a SC pointing request that fulfills the following conditions:

- a SC axis is pointed towards the SC velocity relative to another object.
- the remaining degree of freedom in the SC attitude is determined by a phase angle from the SC orbital pole with respect to the object and another SC axis.
- the two SC axes shall not be parallel.
- the phase angle is the angle in the plane perpendicular to the target direction from the projection of the reference inertial direction to the projection of the SC axis, a positive angle meaning a positive rotation around the target direction. The resulting SC attitude is defined in ANNEX E.

4.11.2 DEFINITION FILE TEMPLATE

4.11.2.1 The following template shall be used to build the definitions for a PRM containing velocity pointing with orbital pole yaw steering requests. The variable content is shown by variable names between % symbols.

```
<metadata>
  <TIME_SYSTEM>UTC</TIME_SYSTEM>
  <definition name="%definitionName%" version="%definitionVersion%" />
    <frame baseFrame="none" name="%inertialFrameName%" />
    <frame baseFrame="%inertialFrameName%" name="%spacecraftFrameName%" />
    <orbit name="%spacecraftName%">
      <!-- OEM containing the SC orbit  -->
      <orbitFile>%OEM%</orbitFile>
    </orbit>
    <orbit name="%targetBodyName%">
      <!-- Object name for the reference target body  -->
      <ephObject>%targetBodyName%</ephObject>
    </orbit>
    <dirVector name="velocity" operator="derivative">
      <origin ref="%targetBodyName%" />
```

```
      <target ref="%spacecraftName%" />
    </dirVector>
    <dirVector name="position">
      <origin ref="%targetBodyName%" />
      <target ref="%spacecraftName%" />
    </dirVector>
    <dirVector name="orbitalPole" operator="cross">
      <dirVector ref="position"/>
      <!-- Coordinates of the satellite velocity -->
      <dirVector ref="velocity"/>
    </dirVector>
    <block name="velocityWithOrbitalPoleYawSteering">
      <startEpoch localName="blockStart" />
      <endEpoch localName="blockEnd" />
      <attitude>
        <!-- Coordinates of default axis to be pointed -->
        <frameDir localName="boresight" />
        <baseFrameDir ref="velocity" />
        <phaseAngle>
          <!-- SC reference direction for phase angle -->
          <frameDir frame="%spacecraftFrameName%"
                    coord="%phaseCoordType%"
                    units="%phaseFrameUnits%">%phaseCoords%</frameDir>
          <!-- Reference direction for phase angle -->
          <baseFrameDir ref="orbitalPole" />
          <projAngle localName="phaseAngle" />
        </phaseAngle>
      </attitude>
    </block>
  </definition>
</metadata>
```

4.11.2.2 The variable content in the definitions template shall be substituted according to the rules in the following table. The values provided in the Tag column are those in the container: /prm/body/segment/metadata/definition/

Variable	Tag	Description	Allowed values	Example value
%definitionName%	@name	The identifier for the pointing elements definition; to be referenced in the generation of requests	-	
%definitionVersion%	@version	Version of the definition	By convention	1.3
%inertialFrameName%	frame[1]/@name frame[2]/@baseframe	Inertial reference frame name.	One of the inertial frames from ANNEX A.	EME2000
%spacecraftFrameName%	frame[2]/@name block/attitude/phaseAngle/frameDir/@frame	SC reference frame name	-	SC

Variable	Tag	Description	Allowed values	Example value
%spacecraftName%	orbit[1]/@name dirVector[1]/target/@ref surfaceVector[2]/traget/@ref	SC name	-	MEX
%OEM%	orbit[1]/orbitFile	The URL to the orbit file containing the satellite trajectory (typically in OEM format)		
%targetBodyName%	orbit[2]/@name dirVector[1]/origin/@ref dirVector[2]/origin/@ref orbit[2]/ephObject	The name of the body to be used as target for the pointing	Value given in [9]	Mars
%phaseCoordType%	block/attitude/phaseAngle/frameDir/@coord	Type of coordinates defining the direction of the phase direction vector in SC frame.	cartesian spherical	cartesian
%phaseFrameUnits%	block/attitude/phaseAngle/frameDir/@units	Units of the phase direction vector in SC reference frame	For %phaseCoordType%=spherical: units="deg" or units="rad" For %phaseCoordType%=cartesian this variable must be an empty string.	deg
%phaseCoords%	block/attitude/phaseAngle/frameDir	The value of the direction vector coordinates in SC frame to compute the phase angle with respect to the base phase coordinates	Any conforming to the direction type in Table 1	0. 1. 0.

4.11.2.3 The direction vector type variables shall be given by its coordinates following the coordinates representation for direction vector type from section 3.3.2.9.

4.11.3 REQUEST BODY TEMPLATE

4.11.3.1 The following template shall be used to build velocity pointing with orbital pole yaw steering request blocks inside the PRM body. The variable content is shown between % symbols.

```
<data>
  <timeline frame="%spacecraftFrameName%">
    <block ref="velocityWithOrbitalPoleYawSteering">
      <!-- Pointing request start time -->
      <blockStart>%blockStartEpoch%</blockStart>
      <!-- Pointing request end time -->
      <blockEnd>%blockEndEpoch%</blockEnd>
      <!-- SC axis to be pointed in the direction of the relative velocity -->
      <boresight frame="%spacecraftFrameName%"
                 coord="%spacecraftCoordType%"
                 units="%spacecraftCoordUnits%">%spacecraftBoreCoords%</boresight>
      <phaseAngle units="%phaseAngleUnits%">%phaseAngle%</phaseAngle>
    </block>
  </timeline>
</data>
```

4.11.3.2 The variable content in the pointing request block template shall be substituted according to the rules in the following table. The values provided in the Tag column are those in the container: `/prm/body/segment/data/timeline/block/`

Variable	Tag	Description	Allowed values	Example value
`%spacecraftFrameName%`	`../@frame` `boresight/@frame`	SC reference frame name	-	`SC`
`%blockStartEpoch%`	`blockStart`	Start epoch of the pointing request	Epoch according to 3.3.2.1	`2009-09-25T19:00:00.`
`%blockEndEpoch%`	`blockEnd`	End epoch of the pointing request	Epoch according to 3.3.2.1	`2009-09-25T20:00:00.`
`%spacecraftCoordType%`	`boresight/@coord`	Coordinate type of the given pointed axis	cartesian spherical	`cartesian`
`%spacecraftCoordUnits%`	`boresight/@units`	Units of the SC main pointing axis	For `%spacecraftCoordType%=spherical`: `units="deg"` or `units="rad"` For `%spacecraftCoordType%=cartesian` this variable must be an empty string.	`deg`

Variable	Tag	Description	Allowed values	Example value
%spacecraftBoreCoords%	boresight	Coordinates of the direction vector in the SC reference frame for the main pointing axis	Any conforming to the direction type in Table 1	0.052336 0. 0.99863
%phaseAngleUnits%	phaseAngle/@units	Units for the phase angle	deg rad	deg
%phaseAngle%	phaseAngle	Angle value according to the real value representation in 3.3.2.6	-	10.

5 RULES FOR THE CONSTRUCTION OF MISSION SPECIFIC PRMS

This section deals with the creation of a PRM from the lower level building elements for those cases not covered by already pre-defined templates in section 4.

5.1 GENERAL RULES

There are two essential elements in the construction of a PRM

5.1.1 Any PRM shall conform to the high level structure defined in section 3.2.

Therefore the first step in building a PRM from scratch is to prepare the structure template to receive the detailed information.

5.1.2 Any PRM shall be built as a collection of elements of the types defined in sections 3.3.2 and 3.3.3.

5.2 PRM HIGH LEVEL STRUCTURE

5.2.1 The PRM shall follow the structure of all other CCSDS navigation messages in their XML representation (See reference [6]).

5.2.2 The generation of a PRM from scratch shall be based on the preparation of the following basic template.

```
<prm id="CCSDS_PRM_VERS" version="1.0">
       <header>
              <CREATION_DATE>2012-281T17:26:06</CREATION_DATE>
              <ORIGINATOR>ESA</ORIGINATOR>
       </header>
       <body>
              &lt;segment&gt;
                     <metadata>
                     </metadata>
                     <data>
                     </data>
              &lt;/segment&gt;
              ...
       </body>
</prm>
```

5.2.3 The `id` attribute in the PRM root element shall be CCSDS_PRM_VERS.

5.2.4 The `version` attribute in the PRM root element shall be 1.0.

5.2.5 The user shall provide values for the `CREATION_DATE` and `ORIGINATOR` elements following the rules in ANNEX A, ANNEX B and ANNEX C.

Note: The detailed structure of the `body` section depends on the actual nature of the PRM being built; details are provided in the following sections.

5.3 PRM SEGMENT

This section focuses in the actual pointing request aspects of the PRM. The PRM `body` contains a number of `segment` elements providein the details of the poiting request. As described in section 3.2, the two main constituents of any PRM are the `definition` element in the `metadata` container and the requests `data` element that contains specific request information and references to the definitions. The most general situation is that in which a segment contains definitions and requests that reference to the definitions in the same or other segments. From the point of view of generality, it is sufficient to describe the process of building definition blocks and request blocks. The referencing mechanism is detailed in sections 3.4.2 and 3.4.3.

5.3.1 The reference structure of a segment shall be according to the following template

```
&lt;segment&gt;
        <metadata>
                <TIME_SYSTEM>UTC</TIME_SYSTEM>
                <definition>
                </definition>
                ...
        </metadata>
        <data>
                <timeline>
                        <block>
                        </block>
                        ...
                </timeline>
                ...
        </data>
&lt;/segment&gt;
```

5.3.1.1 The definition sections shall be built from any combination of the building blocks contained in section 3.3 and only by those building blocks.

5.3.1.2 The attitude block sections shall be built from any combination of the building blocks contained in section 3.3 and only by those building blocks.

5.3.2 DEFINITION SECTION

5.3.2.1 The definition section of a PRM shall be contained in the `metadata` container.

5.3.2.2 Within each `metadata` container there may be one or more definition sections.

5.3.2.3 Each definition start element shall be furnished with a `name` attribute and a `version` attribute for further reference.

```
<definition name="defBlock" version="a.b">
...
</definition>
```

5.3.2.4 The definition section shall identify the reference frames involved in the pointing request.

5.3.2.5 The definition section shall identify the orbital references involved in the pointing request. This can be provided either as ephemeris files, e.g. OEM, or as a common designator of a celestial body, e.g. from DE405.

5.3.2.6 The definition section shall define the privileged directions (normally based on the orbital references defined before).

5.3.2.7 The definition section shall identify attitude blocks to be referenced from the pointing request part (`data` block)

The following scheme provides an example of the construction of the definition section according to the rules provided above.

```
<definition name="defBlock" version="a.b">
       <frame ... name="X" />
       <frame ... name="Y" />
       <orbit name="OEM" />
       <orbit name="EPH" />
       <dirVector>
              <origin ref="OEM" />
              <target ref="EPH" />
       </dirVector>
       <block name="A">
       ...
       </block>
       <block name="B">
       ...
       </block>
</definition>
```

The following paragraphs define the sequence of steps to build a general definition section of a PRM. Because the number of combinations is as wide as the possible definition of attitude elements and their combinations, the steps focus on the construction of a simple PRM definition section example showing possible alternatives when there is more than one way to build a certain element. New elements added in each step of the process are identified with bold-italics type font.

5.3.2.7.1 The base frame shall be defined in the definitions section.

```
<definition name="defBlock" version="a.b">
       <frame baseFrame="none" name="EME2000" />
       ...
</definition>
```

5.3.2.7.2 Every reference frame in the definitions section shall be identified with a unique name.

```
<definition name="defBlock" version="a.b">
       <frame baseFrame="none" name="EME2000" />
       <frame baseFrame="EME2000" name="SC" />
       ...
</definition>
```

5.3.2.8 The definition section shall identify the orbital references that will be used in the pointing request.

5.3.2.8.1 The definitions section shall identify all required spacecraft orbits provided through their ephemeris.

```
<definition name=”defBlock” version=“a.b”>
       <frame baseFrame=”none” name=”EME2000” />
       <frame baseFrame=”EME2000” name=”SC” />
       <orbit name=”SC#1”>
              <orbitFile>SC#1.oem.xml</orbitFile>
       </orbit>
       ...
</definition>
```

5.3.2.8.2 The definitions section shall identify all required celestial bodies trajectories through their common designators (i.e. JPL codes).

```
<definition name=”defBlock” version=“a.b”>
       <frame baseFrame=”none” name=”EME2000” />
       <frame baseFrame=”EME2000” name=”SC” />
       <orbit name=”SC#1”>
              <orbitFile>SC#1.oem.xml</orbitFile>
       </orbit>
       <orbit name=”Jupiter”>
              <ephObject>JUPITER</ephObject>
       </orbit>
       ...
</definition>
```

Note: the contents of the parameter `name` *in the* `orbit` *element is a user provided value. The actual value defining the ephemerides according to* [9] *is the value of the element* `ephObject`.

5.3.2.9 The definitions section shall define all privileged directions needed to define the request.

The objective is to define directions in the spacecraft frame or between two time-evolving objects (e.g. the spacecraft and a celestial body) such that those directions can be referenced later in the request in a generic manner.

```
<definition name=”defBlock” version=“a.b”>
       <frame baseFrame=”none” name=”EME2000” />
       <frame baseFrame=”EME2000” name=”SC” />
       <orbit name=”SC#1”>
              <orbitFile>SC#1.oem.xml</orbitFile>
       </orbit>
       <orbit name=”Jupiter”>
              <ephObject>JUPITER</ephObject>
       </orbit>
       <dirVector name=”targetBody”>
              <origin ref=”SC#1” />
              <target ref=”Jupiter” />
       </dirVector>
       <dirVector name=”boresight” frame=”SC”>0.0 0.0 1.0</dirVector>
       ...
</definition>
```

In the example being constructed two directions are defined

- `targetBody`: that identifies the direction from (`origin`) the spacecraft pointing to (`target`) Jupiter. Each end in the direction is defined by its respective orbit reference.

- `boresight`: direction in the spacecraft body frame.

5.3.2.10 The definitions section shall include the attitude block definition.

```
<definition name="defBlock" version="a.b">
      <frame baseFrame="none" name="EME2000" />
      <frame baseFrame="EME2000" name="SC" />
      <orbit name="SC#1">
            <orbitFile>SC#1.oem.xml</orbitFile>
      </orbit>
      <orbit name="Jupiter">
            <ephObject>JUPITER</ephObject>
      </orbit>
      <dirVector name="targetBody">
            <origin ref="SC#1" />
            <target ref="Jupiter" />
      </dirVector>
      <dirVector name="boresight" frame="SC">0.0 0.0 1.0</dirVector>
      <block name="attBlock">
      ...
      </block>
</definition>
```

5.3.2.10.1 Each attitude block with the definitions section shall provide block start and block end identifiers.

5.3.2.10.2 The block start and block end epochs shall be given unique identifiers for further reference.

The reason not to give actual epochs but unique identfieris is that the defined block is generic and can then be used for any required time interval later in the request part of the PRM.

```
<definition name="defBlock" version="a.b">
      <frame baseFrame="none" name="EME2000" />
      <frame baseFrame="EME2000" name="SC" />
      <orbit name="SC#1">
            <orbitFile>SC#1.oem.xml</orbitFile>
      </orbit>
      <orbit name="Jupiter">
            <ephObject>JUPITER</ephObject>
      </orbit>
      <dirVector name="targetBody">
            <origin ref="SC#1" />
            <target ref="Jupiter" />
      </dirVector>
      <dirVector name="boresight" frame="SC">0.0 0.0 1.0</dirVector>
      <block name="attBlock">
            <startEpoch localName="blockStart" />
            <endEpoch localName="blockEnd" />
            ...
```

```
        </block>
</definition>
```

5.3.2.11 The attitude block shall include the attitude definition section.

```
<definition name="defBlock" version="a.b">
        <frame baseFrame="none" name="EME2000" />
        <frame baseFrame="EME2000" name="SC" />
        <orbit name="SC#1">
                <orbitFile>SC#1.oem.xml</orbitFile>
        </orbit>
        <orbit name="Jupiter">
                <ephObject>JUPITER</ephObject>
        </orbit>
        <dirVector name="targetBody">
                <origin ref="SC#1" />
                <target ref="Jupiter" />
        </dirVector>
        <dirVector name="boresight" frame="SC">0.0 0.0 1.0</dirVector>
        <block name="attBlock">
                <startEpoch localName="blockStart" />
                <endEpoch localName="blockEnd" />
                <attitude>
                ...
                </attitude>
        </block>
</definition>
```

5.3.2.11.1 The attitude definition section shall include the spacecraft axes to be pointed and the targets external to the spacecraft at which to point.

Note: As a general rule the pointing request is provided by means of a spacecraft direction pointing (boresight, defined by an instrument direction, antenna or an arbitrary direction in the spacecraft body frame) to a direction external to the SC (e.g. celestial body) and then a rotation around the boresight to complete the attitude definition for the request (this last can be left undefined or an angular rate around the boresight provided for spin stabilized spacecraft)

5.3.2.11.2 The attitude definition section shall define the main direction of the pointing.

5.3.2.11.3 The definition of the direction to be pointed shall use the reference to the directions resulting from te implementation of steps 5.3.2.11.1 and 5.3.2.11.2.

5.3.2.11.4 The definition of the direction to point at shall use the reference to the directions resulting from te implementation of steps 5.3.2.11.1 and 5.3.2.11.2..

```
<definition name="defBlock" version="a.b">
        <frame baseFrame="none" name="EME2000" />
        <frame baseFrame="EME2000" name="SC" />
        <orbit name="SC#1">
                <orbitFile>SC#1.oem.xml</orbitFile>
        </orbit>
        <orbit name="Jupiter">
                <ephObject>JUPITER</ephObject>
        </orbit>
        <dirVector name="targetBody">
```

```
			<origin ref=”SC#1” />
			<target ref=”Jupiter” />
		</dirVector>
		<dirVector name=”boresight” frame=”SC”>0.0 0.0 1.0</dirVector>
		<block name=”attBlock”>
			<startEpoch localName=”blockStart” />
			<endEpoch localName=”blockEnd” />
			<attitude>
				<frameDir ref=”boresight” />
				<baseFrameDir ref=”targetBody” />
			...
			</attitude>
		</block>
	</definition>
```

The definition provided in the example is such that the request is meant to align the boresight in the direction from the spacecraft to Jupiter. Note that the reference directions had already being defined in step 5.3.2.9, therefore it is now only necessary to refer to them by the provided names (reference to the value in the `localName` attribute through the attribute `ref`)

5.3.2.12 The definitions section shall close any remaining degree of freedom.

There are two ways in which the remaining degree of freedom can be resolved

5.3.2.13 The attitude section may contain the definition of a phase around the spacecraft pointing direction (e.g. boresight).

In this case a direction in the spacecraft body frame that forms a given angle with a direction defined in a frame external to the spacecraft (e.g. inertial frame) defines the phase angle

```
	<definition name=”defBlock” version=“a.b”>
		<frame baseFrame=”none” name=”EME2000” />
		<frame baseFrame=”EME2000” name=”SC” />
		<orbit name=”SC#1”>
			<orbitFile>SC#1.oem.xml</orbitFile>
		</orbit>
		<orbit name=”Jupiter”>
			<ephObject>JUPITER</ephObject>
		</orbit>
		<dirVector name=”targetBody”>
			<origin ref=”SC#1” />
			<target ref=”Jupiter” />
		</dirVector>
		<dirVector name=”boresight” frame=”SC”>0.0 0.0 1.0</dirVector>
		<block name=”attBlock”>
			<startEpoch localName=”blockStart” />
			<endEpoch localName=”blockEnd” />
			<attitude>
				<frameDir ref=”boresight” />
				<baseFrameDir ref=”targetBody” />
				<phaseAngle>
					<frameDir frame=”SC”
						coord=”raDec”
						units=”deg”>0.0 90.0</frameDir>
					<baseFrameDir frame=”EME2000”
						coord=”cart”>1.0 0.0 0.0</baseFrameDir>
```

```
                    <angle units=”deg”>45.0</angle>
                </phaseAngle>
            </attitude>
        </block>
</definition>
```

In this case the direction in the spacecraft body axis defined in right ascension and declination (0.0, 90.0) is to form an angle of 45.0 degrees with the x-axis (1.0, 0.0, 0.0) of the inertial reference frame.

5.3.2.13.1 The attitude definition section may contain the definition of a direction to be contained in a plane.

5.3.2.13.2 The attitude section may define the direction of the plane by defining the direction perpendicular to the plane surface.

The plane normal can be defined in different ways:

Example 1: with a fixed unit vector in inertial space.

```
<dirVector frame=”EME2000”
           name=”pNormal”
           coord=”cart”>0.5 0.8661 0.0</dirVector>
```

Example 2: with the orbit pole. The resulting direction is computed from the cross product of the spacecraft position vector and its velocity computed as the derivative of the position vector.

```
<dirVector name=“pNormal” operator=“cross”>
        <dirVector name=“scToTtargetBody”>
                <origin ref=”SC” />
                <target ref=“Jupiter” />
        </dirVector>
        <dirVector ref=“scToTargetBody” operator=”derivative” />
</dirVector>
```

5.3.2.13.3 The attitude definition section may define the direction in the spacecraft body frame to be contained in the previously defined plane.

```
<definition name=”defBlock” version=“a.b”>
        <frame baseFrame=”none” name=”EME2000” />
        <frame baseFrame=”EME2000” name=”SC” />
        <orbit name=”SC#1”>
                <orbitFile>SC#1.oem.xml</orbitFile>
        </orbit>
        <orbit name=”Jupiter”>
                <ephObject>JUPITER</ephObject>
        </orbit>
        <dirVector name=”targetBody”>
                <origin ref=”SC#1” />
                <target ref=”Jupiter” />
        </dirVector>
        <dirVector name=”boresight” frame=”SC”>0.0 0.0 1.0</dirVector>
        <dirVector frame=”EME2000”
                   name=”pNormal”
```

```
                coord=”cart”>0.5 0.8661 0.0</dirVector>
      <block name=”attBlock”>
            <startEpoch localName=”blockStart” />
            <endEpoch localName=”blockEnd” />
            <attitude>
                  <frameDir ref=”boresight” />
                  <baseFrameDir ref=”targetBody” />
                  <phaseAngle>
                        <frameDir frame=”SC”
                                 coord=”raDec”
                                 units=”deg”>0.0 90.0</frameDir>
                        <baseFrameDir ref=”pNormal” />
                        <projAngle units=”deg”>90.0</projAngle>
                  </phaseAngle>
            </attitude>
      </block>
</definition>
```

In this case the direction in the spacecraft body axis defined in right ascension and declination (0.0, 90.0) is to be computed perpendicular (`projAngle`=90.0) with the `pNormal` defined inertial direction (0.5 0.8661, 0.0).

5.3.2.14 The attitude section may leave the rotation around the boresight undefined.

This is the simple case as it is not necessary to provide any additional information for the pointing request; this leaves the phase angle undefined and the pointing is completed just by aligning the boresight with the selected external direction.

```
<definition name=”defBlock” version=“a.b”>
      <frame baseFrame=”none” name=”EME2000” />
      <frame baseFrame=”EME2000” name=”SC” />
      <orbit name=”SC#1”>
            <orbitFile>SC#1.oem.xml</orbitFile>
      </orbit>
      <orbit name=”Jupiter”>
            <ephObject>JUPITER</ephObject>
      </orbit>
      <dirVector name=”targetBody”>
            <origin ref=”SC#1” />
            <target ref=”Jupiter” />
      </dirVector>
      <dirVector name=”boresight” frame=”SC”>0.0 0.0 1.0</dirVector>
      <block name=”attBlock”>
            <startEpoch localName=”blockStart” />
            <endEpoch localName=”blockEnd” />
            <attitude>
                  <frameDir ref=”boresight” />
                  <baseFrameDir ref=”targetBody” />
            </attitude>
      </block>
</definition>
```

5.3.2.15 The attitude section may leave the rotation around the boresight undefined and provide an angular rate around the aligned axis.

```
<definition name=”defBlock” version=“a.b”>
      <frame baseFrame=”none” name=”EME2000” />
      <frame baseFrame=”EME2000” name=”SC” />
      <orbit name=”SC#1”>
```

```
        <orbitFile>SC#1.oem.xml</orbitFile>
    </orbit>
    <orbit name="Jupiter">
        <ephObject>JUPITER</ephObject>
    </orbit>
    <dirVector name="targetBody">
        <origin ref="SC#1" />
        <target ref="Jupiter" />
    </dirVector>
    <dirVector name="boresight" frame="SC">0.0 0.0 1.0</dirVector>
    <block name="attBlock">
        <startEpoch localName="blockStart" />
        <endEpoch localName="blockEnd" />
        <attitude>
            <frameDir ref="boresight" />
            <baseFrameDir ref="targetBody" />
            <angularRate units="deg/s">0.03</angularRate>
        </attitude>
    </block>
</definition>
```

5.3.3 REQUEST SECTION

5.3.3.1 The request section of a PRM shall be contained in the `data` container.

5.3.3.2 The `data` container shall define a `timeline` section.

5.3.3.3 The `timeline` section shall contain one or more attitude `block` sections.

This structure permits the definition of a sequence of requests by the provision of successive blocks in the timeline to define intervals for the different requests.

The following scheme provides an example of the construction of the request section according to the rules provided above.

```
<data>
    <timeline>
        <block ref="attBlock">
        ...
        </block>
    </timeline>
</data>
```

5.3.3.4 Request section step by step (reference case)

5.3.3.4.1 The pointing request associated to the PRM definition may define one single block in the timeline.

```
<data>
    <timeline>
        <block ref="attBlock">
            <blockStart>2013-10-02T00:00:00</blockStart>
            <blockEnd>2013-10-02T14:30:00</blockEnd>
        </block>
    </timeline>
</data>
```

5.3.3.4.2 The pointing request associated to the PRM definition may define several blocks in the timeline.

```
<data>
      <timeline>
            <block ref="attBlock">
                  <blockStart>2013-10-02T00:00:00</blockStart>
                  <blockEnd>2013-10-02T14:30:00</blockEnd>
            </block>
            <block ref="attBlock">
                  <blockStart>2013-10-03T00:00:00</blockStart>
                  <blockEnd>2013-10-03T14:30:00</blockEnd>
            </block>
            <block ref="attBlock">
                  <blockStart>2013-10-04T00:00:00</blockStart>
                  <blockEnd>2013-10-04T14:30:00</blockEnd>
            </block>
      </timeline>
</data>
```

5.3.3.4.3 Each block in the timeline shall provide its start and end time.

5.3.3.5 Request section step by step (configurable boresight)

5.3.3.5.1 The pointing request associated to the PRM definition may reconfigure the definition section to allow for the the selection of the boresight (direction in the spacecraft body frame).

5.3.3.5.2 The definition section of the PRM shall use the `<frameDir localName="boresight" />` construct to identify the reconfigurable spacecraft pointing axis (boresight).

This permits the dynamic selection of the spacecraft direction without having to modify the definition each time a new request is generated (e.g. need to point different instruments to the same target). Then the definition and request section would be as follows.

```
<metadata>
      <TIME_SYSTEM>UTC</TIME_SYSTEM>
      <definition name="defBlock" version="a.b">
            <frame baseFrame="none" name="EME2000" />
            <frame baseFrame="EME2000" name="SC" />
            <orbit name="SC#1">
                  <orbitFile>SC#1.oem.xml</orbitFile>
            </orbit>
            <orbit name="Jupiter">
                  <ephObject>JUPITER</ephObject>
            </orbit>
            <dirVector name="targetBody">
                  <origin ref="SC#1" />
                  <target ref="Jupiter" />
            </dirVector>
            <block name="attBlock">
                  <startEpoch localName="blockStart" />
                  <endEpoch localName="blockEnd" />
                  <attitude>
```

```
                    <frameDir localName="boresight" />
                    <baseFrameDir ref="targetBody" />
                    ...
                </attitude>
            </block>
        </definition>
</metadata>
```

5.3.3.5.3 The pointing request associated to the PRM definition shall provide the definition of the reconfigurable spacecraft pointing axis.

```
<data>
        <timeline>
            <block ref="attBlock">
                <blockStart>2013-10-02T00:00:00</blockStart>
                <blockEnd>2013-10-02T14:30:00</blockEnd>
                <boresight frame="SC"
                        coord="cart">0.0 0.0 1.0</boresight>
            </block>
        </timeline>
</data>
```

The definition section above already provides the attitude request scheme to align the boresight with the target direction defined by `targetBody`. The request defines the specific direction of the boresight to be pointed and closes the definition of the pointing request.

5.3.3.6 Request section step by step (configurable target)

5.3.3.6.1 The pointing request associated to the PRM definition may reconfigure the definition section to allow for the the selection of the target (direction towards which the boresight should be pointed)

5.3.3.6.2 The definition section of the PRM shall use the `<target localName="target" />` construct to identify the reconfigurable target.

This permits the dynamic selection of the target direction without having to modify the definition each time a new request is generated. Then the definition and request section would be as follows.

```
<metadata>
        <TIME_SYSTEM>UTC</TIME_SYSTEM>
        <definition name="defBlock" version="a.b">
            <frame baseFrame="none" name="EME2000" />
            <frame baseFrame="EME2000" name="SC" />
            <orbit name="SC#1">
                <orbitFile>SC#1.oem.xml</orbitFile>
            </orbit>
            <orbit name="Jupiter">
                <ephObject>JUPITER</ephObject>
            </orbit>
            <orbit name="Saturn">
                <ephObject>SATURN</ephObject>
            </orbit>
            <orbit name="Sun">
                <ephObject>SUN</ephObject>
```

```
                </orbit>
                <dirVector name="boresight" frame="SC">0.0 0.0 1.0</dirVector>
                <block name="attBlock">
                        <startEpoch localName="blockStart" />
                        <endEpoch localName="blockEnd" />
                        <attitude>
                                <frameDir ref="boresight" />
                                <baseFrameDir>
                                        <origin ref="SC#1" />
                                        <target localName="target" />
                                </baseFrameDir>
                                ...
                        </attitude>
                </block>
        </definition>
</metadata>
```

5.3.3.6.3 The pointing request associated to the PRM definition shall provide the definition of the reconfigurable target.

```
<data>
        <timeline>
                <block ref="attBlock">
                        <blockStart>2013-10-02T00:00:00</blockStart>
                        <blockEnd>2013-10-02T14:30:00</blockEnd>
                        <target ref="Jupiter" />
                </block>
                <block ref="attBlock">
                        <blockStart>2013-10-03T00:00:00</blockStart>
                        <blockEnd>2013-10-03T14:30:00</blockEnd>
                        <target ref="Saturn" />
                </block>
                <block ref="attBlock">
                        <blockStart>2013-10-04T00:00:00</blockStart>
                        <blockEnd>2013-10-04T14:30:00</blockEnd>
                        <target ref="Sun" />
                </block>
        </timeline>
</data>
```

The definition section above already provides the attitude request scheme to align the boresight with the target direction defined by `targetBody`. The request defines sequentially the specific targets to point to and closes the definition of the pointing request.

ANNEX A

VALUES FOR TIME SYSTEM AND REFERENCE FRAMES

NORMATIVE

The values in this annex represent the set of acceptable values for the time system and reference frames in the PRM. If exchange partners wish to use different settings, the settings should be documented in the ICD.

A1 TIME SYSTEM

Time System Value	Meaning
GMST	Greenwich Mean Sidereal Time
GPS	Global Positioning System
MET	Mission Elapsed Time (see Note)
MRT	Mission Relative Time (see Note)
SCLK	Spacecraft Clock (receiver) (requires rules for interpretation in ICD)
TAI	International Atomic Time
TCB	Barycentric Coordinate Time
TDB	Barycentric Dynamical Time
TCG	Geocentric Coordinate Time
TT	Terrestrial Time
UT1	Universal Time
UTC	Coordinated Universal Time

NOTE – If MET or MRT is chosen as the TIME_SYSTEM, then the epoch of either the start of the mission for MRT, or of the event for MET, should either be given in a comment in the message or provided in an ICD. The time system for the start of the mission or the event should also be provided in the comment or the ICD. If these values are used for the TIME_SYSTEM, then the times given in the file denote a duration from the mission start or event. However, for clarity, an ICD should be used to fully specify the interpretation of the times if these values are to be used. Note that the time format should only utilize three digit days from the MET or MRT epoch, not months and days of the months.

A2 REFERENCE FRAME

Reference Frame Value	**Meaning**
EME2000	Earth Mean Equator and Equinox of J2000
GCRF	Geocentric Celestial Reference Frame
GRC	Greenwich Rotating Coordinates
ICRF	International Celestial Reference Frame
InstrX	Placeholder for any instrument reference frame
ITRF2000	International Terrestrial Reference Frame 2000
ITRF-93	International Terrestrial Reference Frame 1993
ITRF-97	International Terrestrial Reference Frame 1997
MCI	Mars Centered Inertial
RSW	Another name for ‘Radial, Transverse, Normal’
RTN	Radial, Transverse, Normal
SC	Spacecraft body frame
TDR	True of Date, Rotating
TOD	True of Date
TNW	A local orbital coordinate frame that has the x-axis along the velocity vector, W along the orbital angular momentum vector, and N completes the right handed system.

ANNEX B

ITEMS FOR AN INTERFACE CONTROL DOCUMENT

INFORMATIVE

B1 STANDARD ICD ITEMS

In several places in this document there are references to items which supplement an exchange of pointing request data. These items should be specified in an Interface Control Document (ICD) between participants. The ICD should be jointly produced by both participants in a cross-support involving the transfer of pointing request data. This annex compiles those recommendations into a single section. Although the Pointing Request Message described in this document may at times be used in situations in which participants have not negotiated ICDs, ICDs based on the content specified in this Proposed Standard should be developed and negotiated whenever possible.

The following table will be filled in as document matures.

Item	Section

Item	**Section**

ANNEX C

ACRONYMS AND ABBREVIATIONS

INFORMATIVE

Acronym	**Meaning**
ASCII	American Standard Code for Information Interchange
CCSDS	Consultative Committee for Space Data Systems
CNES	Centre National d'Études Spatiale
ESA	European Space Agency
ICD	Interface Control Document
ICRF	International Celestial Reference Frame
ID	identifier
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
ITRF	International Terrestrial Reference Frame
ITRS	International Terrestrial Reference System
JAXA	Japan Aerospace Exploration Agency
JPL	Jet Propulsion Laboratory
KVN	Keyword = Value notation
MOIMS	Mission Operations and Information Management Services
N/A	Not Applicable / Not Available
NASA	National Aeronautics and Space Administration
OEM	Orbit Ephemeris Message
PRM	Pointing Request Message
UTC	Universal Time Coordinated
XML	Extensible Markup Language

ANNEX D

PRM SAMPLES

INFORMATIVE

This annex provides examples that can be constructed following the procedure described in section 5. As per this procedure, the PRM examples are provided as a Deniniton construct in <metadata> followed by the Request construct in <data>.

- EXAMPLE 1: ANTENNA OF SC1 TO SC2

This example shows a pointing request where the Z axis of one spacecraft (SC1) points to another spacecraft (SC2). The names of both SC orbit files act as parameters.

Definition

```
<metadata>
  <TIME_SYSTEM>UTC</TIME_SYSTEM>
  <definition>
    <frame baseFrame="none" name="EME2000" />
    <frame baseFrame="EME2000" name="SC1" />
    <block name="SC2">
      <startEpoch localName="blockStart" />
      <endEpoch localName="blockEnd" />
      <attitude>
        <frameDir frame="SC1">0. 0. 1.</frameDir>
        <baseFrameDir>
          <origin>
            <orbitFile localName="trajectory1">sc1.oem</orbitFile>
          </origin>
          <target>
            <orbitFile localName="trajectory2">sc2.oem</orbitFile>
          </target>
        </baseFrameDir>
        <phaseAngle>
          <frameDir frame="SC1">0. 1. 0.</frameDir>
          <baseFrameDir frame="EME2000">0. 1. 0.</baseFrameDir>
          <projAngle frame="deg">0.</projAngle>
        </phaseAngle>
      </attitude>
    </block>
  </definition>
</metadata>
```

Request

```
<data>
```

```
  <timeline frame="SC1">
    <block ref="SC2">
      <blockStart>2009-09-25T19:00:00.</blockStart>
      <blockEnd>2009-09-25T20:00:00. </blockEnd>
      <trajectory1 sc='SC1'>OEM_0001.SC1</trajectory1>
      <trajectory2 sc='SC2'>OEM_0002.SC2</trajectory2>
    </block>
  </timeline>
</data
```

- EXAMPLE 2: SIMPLE RASTER

This example shows how to construct a simple raster with 2 points.

The SC attitude is constructed by applying a rotation around the SC Y axis relative to the basic inertial pointing attitude from example 1.

The rotation angle versus time is defined by interpolation of a table that specifies rotation angles and angular rates at certain times. The interpolation in each time interval asumes that it is done by means of a polynomial of degree 3 defined by the rotation angle and rate at the border of each time interval. The example results in a raster with two points at 0 and 10 degree from the target connected by a slew that is continuous in rotation angle and rate. The rotation angle versus time is shown in Figure 1.

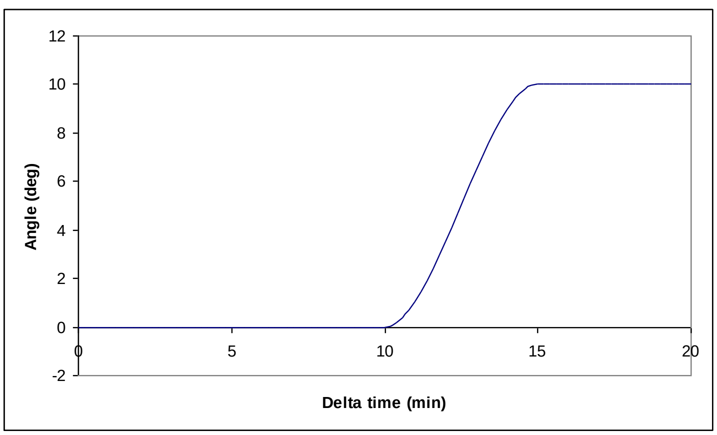


Figure 1 Rotation angle versus time

Definition

```
<metadata>
  <TIME_SYSTEM>UTC</TIME_SYSTEM>
  <definition>
    <frame baseFrame="none" name="EME2000" />
    <frame baseFrame="EME2000" name="SC" />
    <orbit name="SC">
      <orbitFile> sc.oem </orbitFile>
    </orbit>
```

```
    <orbit name=“Sun”>
      <ephObject>SUN</ephObject>
    </orbit>
    <dirVector name="DefaultSunPerpendicular" frame="SC">0. 1. 0.</dirVector>
    <dirVector name="DefaultBoresight" frame="SC">0. 0. 1.</dirVector>
    <dirVector name=“Sun”>
      <origin ref="SC"/>
      <target ref=“Sun”/>
    </dirVector>
    <phaseAngle name=“Sun”>
      <frameDir localName="SunPerpendicular" ref="DefaultSunPerpendicular" />
      <baseFrameDir ref=“Sun”/>
      <angle units=“deg”>90.</angle>
    </phaseAngle>
    <!-- Inertial block modified to allow Raster rotation parameter -->
    <block name="Inertial">
      <startEpoch localName="blockStart" />
      <endEpoch localName="blockEnd" />
      <attitude>
        <attitude>
          <frameDir localName="Boresight" ref="DefaultBoresight" />
          <baseFrameDir localName="target" />
          <phaseAngle ref=“Sun” localName="Roll" />
        </attitude>
        <rotation localName="raster">0. 0. 0. 1.</rotation>
      </attitude>
    </block>
    <!-- Definition of boresights and targets -->
    <dirVector name="Boresight2"
             coord="Spherical"
             frame="SC" units=“deg”>2.5 89.</dirVector>
    <dirVector name="Vega"
             coord="raDec"
             frame="EME2000">18:36:56.336 38:47:01.18</dirVector>
    <!-- Definition of SimpleRaster -->
    <rotation name="SimpleRaster">
      <axis frame="SC">0. 1. 0.</axis>
      <angle>
        <epochList>
          <refEpoch>2009-09-25T19:00:00.</refEpoch>
          <durationList units="min">0. 10. 15. 20.</durationList>
        </epochList>
        <valueList units=“deg”>0. 0. 10. 10.</valueList>
        <derivativeList units="deg/min">0. 0. 0. 0.</derivativeList>
      </angle>
    </rotation>
```

	`` </definition> </metadata> ```
Request	``` <data> <timeline frame="SC"> <block ref="Inertial"> <blockStart> 2009-09-25T19:00:00. </blockStart> <blockEnd> 2009-09-25T20:00:00. </blockEnd> <boresight ref="Boresight2" /> <target ref="Vega" /> <raster ref="SimpleRaster" /> </block> </timeline> </data> ```

ANNEX E

ATTITUDE AND FRAMES CONVENTIONS

INFORMATIVE

Different attitude representations are used to describe the attitude of a reference frame with respect to another (that is referred to as its base frame). The transformation from the base frame to the derived frame (frame being described) can be defined in several ways. The adopted conventions are defined here.

1. Quaternion. The transformation from base frame to the derived frame is defined as follows.

   If q is the normalized attitude quaternion, the attitude matrix of derived frame with respect to base frame is

$$M = \begin{pmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{pmatrix}$$

   i.e. a component row of a vector expressed with respect to the base frame $v_{\text{base frame}}$ corresponds to the component row

$$v_{\text{derived frame}} = M\, v_{\text{base frame}}$$

   in the derived frame.

   Example of reference frame defined by attitude quaternion:

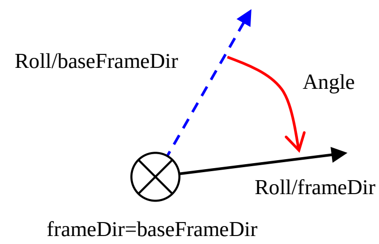
```
<frame name='Instrument1' baseFrame='SC'>
  <attitude>
    <rotation> 0. 0. 0. 1. </rotation>
  </attitude>
</frame>
```

2. Pointing direction and `phaseAngle`. Two vectors relative to the derived frame are given by the `frameDir` elements. Two vectors relative to the base frame are given by the `baseFrameDir` elements. The derived frame attitude results from aligning the direction vector defined by `attitude`-child element `frameDir` in the derived frame and by the `baseFrameDir`-element in the base frame. This direction is in the following referred to as pointing direction. The degree of freedom around the pointing direction is determined by the `phaseAngle` element. Two alternatives are considered for the roll definition: providing an angle between two directions (`angle` element), or providing the angle between their projections in the plane perpendicular to the pointing direction (`projAngle` element).

   For the roll option defined with `angle` geometrically there may be one, two or no solutions fulfiling the requirements that the pointing direction is maintained and the two axes provided in the phaseAngle element have the required angle. If there are two solutions, the solution selected is the one such that the axis defined by the `phaseAngle` child element `frameDir` has a positive projection on the cross product of pointing direction and the `phaseAngle` child element `baseFrameDir`. The discarded solution can be selected by selecting the opposite direction for one of the axes defined in the `phaseAngle`-element and replacing the contents of the `angle` element by its 180deg complement. If there is no solution, the axes are put such that they form an angle as close as possible to the requested angle.

Example of reference frame defined by pointing direction and Roll:

```
<frame name='Instrument1' baseFrame='SC'>
  <attitude>
    <frameDir frame='Instrument1'> 1. 0. 0. </frameDir>
    <baseFrameDir frame='SC'> 1. 0. 0. </baseFrameDir>
    <phaseAngle>
      <frameDir frame='Instrument1'> 0. 1. 0. </frameDir>
      <baseFrameDir frame='SC'> 0. 0. 1. </baseFrameDir>
      <angle units='deg'> 0. </angle>
    </phaseAngle>
  </attitude>
</frame>
```

	
<phaseAngle> <frameDir …> <baseFrameDir …> <angle units='deg'> 45. </angle> </phaseAngle>	<phaseAngle> <frameDir …> <baseFrameDir …> <projAngle units='deg'> 45. </projAngle> </phaseAngle>
Roll angle: angle between a derived frame direction and a base frame direction.	Roll projected angle: angle from a base frame direction to a derived frame direction.

Rotation. A `Rotation` element defines a rotation in terms of a rotation axis $\vec{e}_{rot}$ defined by means of the `rotAxis` element and a rotation angle α. If the rotation is applied to a direction vector $\vec{e}$ the resulting direction vector $\vec{e}'$ is obtained by a right handed rotation of the direction vector around the rotation axis i.e. $\vec{e}' = (\vec{e}\cdot\vec{e}_{rot})\vec{e}_{rot} + \cos(\alpha)((\vec{e} - (\vec{e}\cdot\vec{e}_{rot})\vec{e}_{rot}) + \sin(\alpha)(\vec{e}_{rot}\times\vec{e})$.
If the rotation is applied to a derived attitude the resulting derived attitude frame is defined by performing a right handed rotation of each basis vector around the rotation axis.

If there is more than one `Rotation` element present the rotations are applied in order of appearance in the file.

FRAMES.

A `frame` element defines a reference frame. All frames are defined with respect to another frame (designated as its "*base frame*"). A *base frame* can be either the *root frame* or another *independent frame*.

The *root frame* is the root of the tree formed by all frames defined in a PRM.

The *root frame* is the only frame that has no *base frame*.

Only one *root frame* is allowed per PRM.

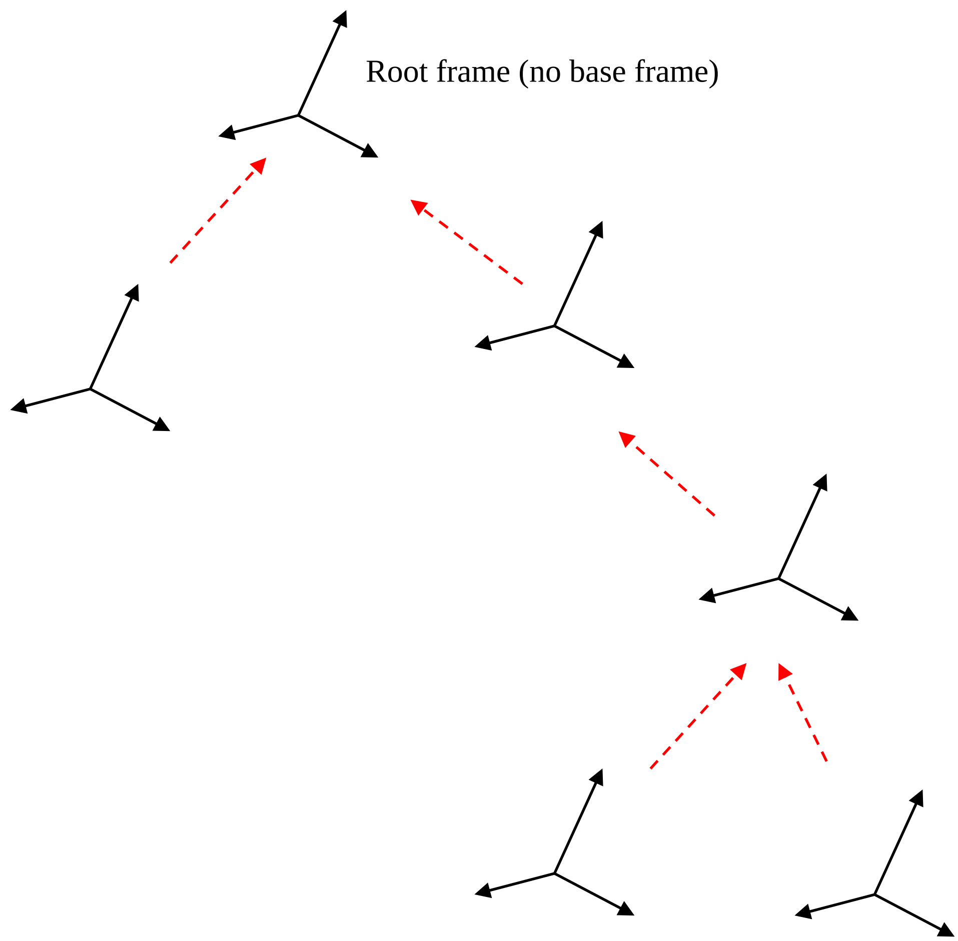


Figure 2 Example tree of PRM frames. Dashed arrows connect every frame to its 'base frame'.

ANNEX F

LIST OPERATORS

INFORMATIVE

In the following the ***List of Reals*** instances constructed by use of the `operator` attribute is defined.

1. `join`: allows to have two or more child lists of type ***List of Reals***. All child lists must have the same unit type. The resulting list appends the child lists in order of appearance. It has the same unit type as the child lists.

List element	**Resulting list**
<realList operator='join'> <realList> 1. 2. 3. </realList> <realList> 4. 5. </realList> </realList>	1. 2. 3. 4. 5.

2. `plus`: allows to have two or more child lists of type ***List of Reals***. All child lists must have the same lengths and unit type. The resulting list is obtained by adding the corresponding components of the child lists. It has the same unit type as the child lists.

List element	**Resulting list**
<realList operator='plus'> <realList> 1. 2. 3. </realList> <realList> 4. 5. 6. </realList> </realList>	5. 7. 9.

3. `minus`: allows to have two child lists of type ***List of Reals***. The child lists must have the same lengths and unit type. The resulting list is obtained by subtracting the corresponding components of the child lists. It has the same unit type as the child lists.

List element	**Resulting list**
<realList operator='minus'> <realList> 1. 2. 3. </realList> <realList> 4. 5. 6. </realList> </realList>	-3. -3. -3.

4. unaryMinus: allows to have a child list of type ***List of Reals***. The resulting list is obtained by sign change of the corresponding components of the child list. It has the same unit type as the child lists.

List element	**Resulting list**
<realList operator='unaryMinus'> <realList> 1. 2. 3. </realList> </realList>	-1. -2. -3.

5. multiply: allows to have two or more child elements of type ***List of Reals*** or ***Real***. All child elements of type ***List of Reals*** must have the same length. The resulting list is obtained by multiplying the corresponding components of the child lists and multiplying the resulting list with each ***Real***. It has the unit corresponding to the product of units of all children.

List element	**Resulting list**
<realList operator='multiply'> <realList units='m'> 4. 5. 6. </realList> <realList units='m'> 1. 2. 3. </realList> <real units='m'> 2. </real> </realList>	8. 20. 36. the resulting unit is m**3
<realList operator='multiply'> <realList units='m'> 4. 5. 6. </realList> <realList units='m'> 2. 3. 5. </realList> <realList units='m'> 1. 2. 3. </realList> </realList>	8. 30. 90. the resulting unit is m**3

6. divide: allows to have two child lists of type ***List of Reals***; the second child can be of type ***Real***. The child lists must have the same lengths. The resulting list is obtained by dividing the components of the first ***List or Reals*** by the components of the second ***List of Reals*** or by the ***Real***. The resulting unit is given by the quotient of units of the child elements.

List element	**Resulting list**
<realList operator='divide'> <realList units='deg'> 2. 4. 6. </realList> <realList units='s'>1. 2. 3.</realList> </realList>	2. 2. 2. the resulting unit is deg/s

List element	Resulting list
<realList operator='divide'> <realList units='deg'> 2. 4. 6. </realList> <real units='s'> 2. </real> </realList>	1. 2. 3. the resulting unit is deg/s

7. take: The resulting list contains a subset of the elements of the child list, corresponding to the components of the child list starting in the firstIndex and until the lastIndex. If no firstIndex is provided the first component will be the first taken, if no lastIndex is provided then the end of the child list will be reached. The resulting list has the same units as the child list.

List element	Resulting list
<realList operator='take'> <realList> 1. 2. 3. </realList> <firstIndex> 2 </firstIndex> <lastIndex> 2 </lastIndex> </realList>	2.

8. repeat: The resulting list contains the child list repeated a certain number of times that are given from the Integer type element nTimes. It has the same units as the child list.

List element	Resulting list
<realList operator='repeat'> <realList> 1. 2. 3. </realList> <nTimes> 2 </nTimes> </realList>	1. 2. 3. 1. 2. 3.

9. cumm: The resulting list is built from a single child list. The first component of the resulting list is the first component of the child list. From that component on, the component n of the resulting list is computed as the component n-1 of the resulting list plus the component n of the child list. The resulting list has the same units as the child list.

List element	Resulting list
<realList operator='cumm'> <realList> 1. 2. 3. </realList> </realList>	1. 3. 6.

10. derivative: allows to have two child lists of type ***List of Reals***. The child lists must have the same lengths and unit type. The fist list contains the list of values to be derived; the scond list contains the independent variable to be used nn the derivation. The resulting list is obtained by implementation of the mathematical derivation operator of the first list with respect to the second. The resulting list has the units of the first child list over the units of the second child list. The resulting list may have different size than the input lists depending on the derivation algorithm applied.

List element	**Resulting list**
<realist operator='derivative'> <realist units='deg'> 1. 4. 8. </realList> <realist units='s'> 1. 2. 3. </realList> </realList>	2.0 4.0 Linear derivation used. The resulting units are deg/s.

ANNEX G

SUPPORTED UNITS

INFORMATIVE

The `units` attribute reports the units in which a value for a physical variable is provided. The following table lists the unit types, possible values and adopted default value per unit type (not exhaustive).

Unit Type	**Default value**	**Allowed values**	**Description**
None	`none`	`none`	Dimensionless
Angle	`deg`	`deg`	Degrees
		`rad`	Radians
		`arcMin`	Arcminutes
		`arcSec`	Arcseconds
Angular velocity	`deg/sec`	`deg/s`	Degrees per second
		`deg/min`	Degrees per minute
		`rad/s`	Radians per second
		`arcSec/s`	Arc seconds per second
Distance	`km`	`km`	Kilometers
		`m`	Meters
Duration	`s`	`s`	Seconds
		`m`	Minutes
		`h`	Hours
		`d`	Days
		`dhms`	Duration specified in calendar format ([+-][[[dddT]hh:]mm:]ss[.ss]

ANNEX H

SECURITY, SANA, AND PATENT CONSIDERATIONS

(INFORMATIVE)

H1 SECURITY CONSIDERATIONS

H1.1 ANALYSIS OF SECURITY CONSIDERATIONS

This subsection presents the results of an analysis of security considerations applied to the technologies specified in this Recommended Standard.

H1.2 CONSEQUENCES OF NOT APPLYING SECURITY TO THE TECHNOLOGY

The consequences of not applying security to the systems and networks on which this Recommended Standard is implemented could include potential loss, corruption, and theft of data. Because these messages are used in pointing request and potential satellite and instrument pointing maneuvers, the consequences of not applying security to the systems and networks on which this Recommended Standard is implemented could include compromise or loss of the mission if malicious tampering of a particularly severe nature occurs.

H1.3 POTENTIAL THREATS AND ATTACK SCENARIOS

Potential threats or attack scenarios include, but are not limited to, (a) unauthorized access to the programs/processes that generate and interpret the messages, and (b) unauthorized access to the messages during transmission between exchange partners. Protection from unauthorized access during transmission is especially important if the mission utilizes open ground networks, such as the Internet, to provide ground-station connectivity for the exchange of data formatted in compliance with this Recommended Standard. It is strongly recommended that potential threats or attack scenarios applicable to the systems and networks on which this Recommended Standard is implemented be addressed by the management of those systems and networks.

H1.4 DATA PRIVACY

Privacy of data formatted in compliance with the specifications of this Recommended Standard should be assured by the systems and networks on which this Recommended Standard is implemented.

H1.5 DATA INTEGRITY

Integrity of data formatted in compliance with the specifications of this Recommended Standard should be assured by the systems and networks on which this Recommended Standard is implemented.

H1.6 AUTHENTICATION OF COMMUNICATING ENTITIES

Authentication of communicating entities involved in the transport of data which complies with the specifications of this Recommended Standard should be provided by the systems and networks on which this Recommended Standard is implemented.

H1.7 DATA TRANSFER BETWEEN COMMUNICATING ENTITIES

The transfer of data formatted in compliance with this Recommended Standard between communicating entities should be accomplished via secure mechanisms approved by the Information Technology Security functionaries of exchange participants.

H1.8 CONTROL OF ACCESS TO RESOURCES

Control of access to resources should be managed by the systems upon which originator formatting and recipient processing are performed.

H1.9 AUDITING OF RESOURCE USAGE

Auditing of resource usage should be handled by the management of systems and networks on which this Recommended Standard is implemented.

H1.10 UNAUTHORIZED ACCESS

Unauthorized access to the programs/processes that generate and interpret the messages should be prohibited in order to minimize potential threats and attack scenarios.

H1.11 DATA SECURITY IMPLEMENTATION SPECIFICS

Specific information-security interoperability provisions that may apply between agencies and other independent users involved in an exchange of data formatted in compliance with this Recommended Standard could be specified in an ICD.

H2 SANA CONSIDERATIONS

The following PRM related items will be registered with the SANA Operator. The registration rule for new entries in the registry is the approval of new requests by the CCSDS

Navigation Working Group chair. New requests for this registry should be sent to SANA (mailto:info@sanaregistry.org).

- The PRM XML schemas for the PRM templates;

H3 PATENT CONSIDERATIONS

The recommendations of this document have no patent issues.