The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

POINTING REQUEST MESSAGE

RECOMMENDED STANDARD

CCSDS 509.0-B-1

BLUE BOOK

February 2018

The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

POINTING REQUEST MESSAGE

RECOMMENDED STANDARD

CCSDS 509.0-B-1

BLUE BOOK

February 2018

AUTHORITY

Issue:	Recommended Standard, Issue 1
Date:	February 2018
Location:	Washington, DC, USA

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 *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4), and the record of Agency participation in the authorization of this document can be obtained from the CCSDS Secretariat at the e-mail address below.

This document is published and maintained by:

CCSDS Secretariat
National Aeronautics and Space Administration
Washington, DC, USA
E-mail: secretariat@mailman.ccsds.org

STATEMENT OF INTENT

The Consultative Committee for Space Data Systems (CCSDS) is an organization officially established by the management of its members. The Committee meets periodically to address data systems problems that are common to all participants, and to formulate sound technical solutions to these problems. Inasmuch as participation in the CCSDS is completely voluntary, the results of Committee actions are termed **Recommended Standards** and are not considered binding on any Agency.

This **Recommended Standard** is issued by, and represents the consensus of, the CCSDS members. Endorsement of this **Recommendation** is entirely voluntary. Endorsement, however, indicates the following understandings:

- o Whenever a member establishes a CCSDS-related **standard**, this **standard** will be in accord with the relevant **Recommended Standard**. Establishing such a **standard** does not preclude other provisions which a member may develop.
- o Whenever a member establishes a CCSDS-related **standard**, that member will provide other CCSDS members with the following information:
  - -- The **standard** itself.
  - -- The anticipated date of initial operational capability.
  - -- The anticipated duration of operational service.
- o Specific service arrangements shall be made via memoranda of agreement. Neither this **Recommended Standard** nor any ensuing **standard** is a substitute for a memorandum of agreement.

No later than five years from its date of issuance, this **Recommended Standard** will be reviewed by the CCSDS to determine whether it should: (1) remain in effect without change; (2) be changed to reflect the impact of new technologies, new requirements, or new directions; or (3) be retired or canceled.

In those instances when a new version of a **Recommended Standard** is issued, existing CCSDS-related member standards and implementations are not negated or deemed to be non-CCSDS compatible. It is the responsibility of each member to determine when such standards or implementations are to be modified. Each member is, however, strongly encouraged to direct planning for its new standards and implementations towards the later version of the Recommended Standard.

FOREWORD

This document is a Recommended Standard that has been prepared by the Consultative Committee for Space Data Systems (CCSDS). The message described in this Recommended Standard establishes a common framework and provides a common format for the interchange of data describing the request for pointing of a spacecraft. The Recommended 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 Recommended Standard and may incorporate features not addressed by this Recommended Standard.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CCSDS has processes for identifying patent issues and for securing from the patent holder agreement that all licensing policies are reasonable and non-discriminatory. However, CCSDS does not have a patent law staff, and CCSDS shall not be held responsible for identifying any or all such patent rights.

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Recommended Standard is therefore subject to CCSDS document management and change control procedures, which are defined in *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4). 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 sent to the CCSDS Secretariat at the e-mail 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.
– 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 (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.
– UK Space Agency/United Kingdom.

Observer Agencies

– Austrian Space Agency (ASA)/Austria.
– Belgian Federal Science Policy Office (BFSPO)/Belgium.
– Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
– China Satellite Launch and Tracking Control General, Beijing Institute of Tracking and Telecommunications Technology (CLTC/BITTT)/China.
– 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.
– Departamento de Ciência e Tecnologia Aeroespacial (DCTA)/Brazil.
– Electronics and Telecommunications Research Institute (ETRI)/Korea.
– European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
– European Telecommunications Satellite Organization (EUTELSAT)/Europe.
– Geo-Informatics and Space Technology Development Agency (GISTDA)/Thailand.
– Hellenic National Space Committee (HNSC)/Greece.
– Indian Space Research Organization (ISRO)/India.
– Institute of Space Research (IKI)/Russian Federation.
– Korea Aerospace Research Institute (KARI)/Korea.
– Ministry of Communications (MOC)/Israel.
– Mohammed Bin Rashid Space Centre (MBRSC)/United Arab Emirates.
– National Institute of Information and Communications Technology (NICT)/Japan.
– National Oceanic and Atmospheric Administration (NOAA)/USA.
– National Space Agency of the Republic of Kazakhstan (NSARK)/Kazakhstan.
– National Space Organization (NSPO)/Chinese Taipei.
– Naval Center for Space Technology (NCST)/USA.
– Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
– Scientific and Technological Research Council of Turkey (TUBITAK)/Turkey.
– South African National Space Agency (SANSA)/Republic of South Africa.
– Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
– Swedish Space Corporation (SSC)/Sweden.
– Swiss Space Office (SSO)/Switzerland.
– United States Geological Survey (USGS)/USA.

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 509.0-B-1	Pointing Request Message, Recommended Standard, Issue 1	February 2018	Original issue

CONTENTS

CONTENTS (continued)

CONTENTS (continued)

CONTENTS (continued)

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 to the attitude of the spacecraft or to an articulated spacecraft component.

1.2 SCOPE

This Recommended Standard is applicable only to the message format and content, 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 Recommended Standard 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:

– Subsection 3.1 provides a general introduction to the PRM structure.

– Subsection 3.2 provides an overview of the PRM structure.

– Subsection 3.3 specifies the XML elements available for constructing PRMs.

– Subsection 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 implementation conformance statement proforma.

Annex B provides the list of time systems and reference frames used.

Annex C provides details on the use of operators.

Annex D lists supported units.

Annex E discusses security, SANA, and patent considerations for the Pointing Request Message.

Annex F specifies adopted attitude conventions.

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

Annex H provides a list of acronyms and abbreviations used in the Recommended Standard.

Annex I provides sample Pointing Request Messages.

Annex J provides sample instantiation of the PRM templates.

1.6 NOMENCLATURE

1.6.1 NORMATIVE TEXT

The following conventions apply for the normative specifications in this Recommended Standard:

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.

NOTE – These conventions do not imply constraints on diction in text that is clearly informative in nature.

1.6.2 INFORMATIVE TEXT

In the normative sections of this document, informative text is set off from the normative specifications either in notes or under one of the following subsection headings:

– Overview;

– Background;

– Rationale;

– Discussion.

1.7 REFERENCES

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

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

[2] “JPL Solar System Dynamics.” Jet Propulsion Laboratory. http://ssd.jpl.nasa.gov/.

[3] *IEEE Standard for Floating-Point Arithmetic*. 2nd ed. IEEE Std. 754-2008. New York: IEEE, 2008.

[4] Tim Bray, et al., eds. *Extensible Markup Language (XML) 1.0*. 5th ed. W3C Recommendation. N.p.: W3C, 26 November 2008.

[5] Jonathan Marsh, David Orchard, and Daniel Veillard, eds. *XML Inclusions (XInclude) Version 1.0*. 2nd ed. W3C Recommendation. N.p.: W3C, 15 November 2006.

[6] *XML Specification for Navigation Data Messages*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 505.0-B-1. Washington, D.C.: CCSDS, December 2010.

[7] *Orbit Data Messages*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 502.0-B-2. Washington, D.C.: CCSDS, November 2009.

[8] *Attitude Data Messages*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 504.0-B-1. Washington, D.C.: CCSDS, May 2008.

[9] “NAIF Integer ID Codes.” Navigation and Ancillary Information Facility (NAIF). National Aeronautics and Space Administration. http://naif.jpl.nasa.gov/pub/naif/toolkit_docs/FORTRAN/req/naif_ids.html.

[10] “Spacecraft Identifiers.” Space Assigned Numbers Authority. http://sanaregistry.org/r/spacecraftid/.

[11] “Organizations.” Space Assigned Numbers Authority (SANA). http://sanaregistry.org/r/organizations.

[12] CCSDS common module (/ndmxml-<release>-ccsds-common.xsd). “Navigation Data Messages XML Schema.” Space Assigned Numbers Authority (SANA). https://sanaregistry.org/r/ndmxml.

[13] Navigation WG common module (/ndmxml-<release>-navwg-common.xsd). “Navigation Data Messages XML Schema.” Space Assigned Numbers Authority (SANA). https://sanaregistry.org/r/ndmxml.

[14] “Pointing Request Message.” Space Assigned Numbers Authority (SANA). https://sanaregistry.org/r/pointing_request_message.

2 OVERVIEW

2.1 GENERAL

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, a sensor, 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. It is to be noted that the purpose of the PRM is to provide specific requests on the pointing to be attained (element A to be pointed to target B at certain point in time for a certain duration). Not in the scope of the PRM is the process required to achieve such request, e.g.,

– specific mission constraints to be fulfilled (and that may even make the request invalid);

– corrections to the pointing to be applied (e.g., aberration, atmospheric refraction, signal delay, …) to fulfil the request, which are in general mission dependent;

– removal of ambiguities associated to various possible pointing solutions in a request.

PRMs 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. Historically, this information has been transmitted in common language or in various fixed file formats. The PRM offers a formal language alternative.

2.2 POINTING REQUESTS IN SCIENCE OPERATIONS

Pointing requests are transmitted, for instance, from scientists who operate an onboard instrument to the operator of the respective spacecraft. These data transmissions could be inter-agency, e.g., in the 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 through the planet’s 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.

2.3 POINTING REQUESTS IN RELAY OPERATIONS

Pointing requests are passed, for instance, from the user of a relay service to the provider. Examples are:

– 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.

2.4 IMPLEMENTATION BASICS

The PRM is implemented as an XML document only. The complexity of the pointing requests and the involved elements make it necessary to provide an implementation that supports that complexity. XML is a suitable and interoperable approach for structuring the pointing requests in a flexible 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 is desirable but not strictly necessary to understand the PRM principles.

The PRM is implemented as a hierarchical structure of data elements. One of the main principles in the design of the PRM is the ability to create basic entities, so-called definitions that can be aggregated into more complex structures and operations or other definitions. These definitions can be referenced throughout the PRM to allow the systematic and consistent reuse of the defined data structures. The PRM also implements a general inclusion mechanism that permits the generation of reusable definitions of entities that can be further reused in other PRMs.

3 POINTING REQUEST MESSAGE

3.1 OVERVIEW

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 that 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 3.2, 3.3, and section 5) 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 3.3.

3.2 PRM STRUCTURE

3.2.1 STRUCTURE

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 XML PRM structure shall comply with the element characterization in annex subsection A2.1.5

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

3.2.1.5 The XML standard Navigation Data Messages (NDM) header as described in the NDM/XML (see reference [6], section 4) shall follow the `<prm>` tag.

3.2.1.6 The XML version, root element tag, and NDM/XML header shall be constructed as described in the NDM/XML (reference [6], section 4).

3.2.1.7 The final attributes of the `<prm>` tag shall be ‘id’ and ‘version’.

3.2.1.8 The ‘`id`’ attribute shall be ‘`id="CCSDS_PRM_VERS"`’.

3.2.1.9 The ‘`version`’ attribute for the `<prm>` shall be ‘`version="1.0"`’.

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

3.2.1.11 The `<header>` element shall include, at a minimum, the `<CREATION_DATE>` and the `<ORIGINATOR>` of the data.

3.2.1.12 The `<ORIGINATOR>` and associated point of contact should be an organization registered in the SANA organization registry (reference [11]), but this is not required.

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

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

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

3.2.1.16 The `<definition>` elements shall describe data entities that can be fully defined independently of the rest of the PRM.

3.2.1.17 The `<metadata>` element shall contain the `<TIME_SYSTEM>` child to define the reference time scale for the segment. Use of values other than those in annex B shall be documented in an ICD.

3.2.1.18 The `<metadata>` element shall contain the `<OBJECT_ID>` and the `<OBJECT_NAME>` children to define the satellite for which the message is applicable.

3.2.1.19 The `<metadata>` element shall contain the `<START_TIME>` and the `<STOP_TIME>` children to identify the aggregated timelines within the segment.

NOTE – The XML layout shown in figure 3-1 corresponds to the intended PRM structure according to the previous requirements. Lower-level details, as those shown in this layout, are described in subsequent sections of the document. The purpose of the figure is to provide the high-level structure of the PRM document only. The variable content, (to be filled when instantiating the templates) is shown by variable names between % symbols.

3.2.1.20 The PRM shall be contained in a file whose naming scheme should be agreed to on a case-by-case basis between the participating agencies and documented in an ICD.

3.2.1.21 The method of exchanging PRM files should be decided on a case-by-case basis by the participating agencies and documented in an ICD.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<prm id="CCSDS_PRM_VERS" version="1.0">

  <!– Standard CCSDS header -->
  <header>
    <CREATION_DATE>%CREATION_DATE%</CREATION_DATE>
    <ORIGINATOR>%ORIGINATOR%</ORIGINATOR>
  </header>

  <!-- Standard CCSDS body structure -->
  <body>

    <!-- Definition segment -->
    <!-- In this case the segment provides definitions only; no requests -->
    &lt;segment&gt;

      <!-- Medadata block provides reference data and pointing definitions -->
      <metadata>
        <OBJECT_NAME>HIPPARCOS</OBJECT_NAME>
        <OBJECT_ID>090154</OBJECT_ID>
        <TIME_SYSTEM>UTC</TIME_SYSTEM>
        <START_TIME>2018-07-02T00:00:00.000</START_TIME>
        <END_TIME>2018-07-04T00:00:00.000</END_TIME>

        <!-- Definition blocks to be referenced in the data sections -->
        <!-- These are built either from templates in section 4 or -->
        <!-- constructed according to the steps in 5.4.3 -->
        <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>

      <!--- Empty (or absent) data block; no requests in this segment -->
      <data />

    &lt;/segment&gt;

    <!-- First pointing request in this segment-->
    &lt;segment&gt;

      <!-- Medadata block provides reference data and pointing definitions -->
      <metadata>
        <!-- Definitions for the first request -->
        <!-- Not all necessarily referenced later -->
        <OBJECT_NAME>HIPPARCOS</OBJECT_NAME>
        <OBJECT_ID>090154</OBJECT_ID>
        <TIME_SYSTEM>UTC</TIME_SYSTEM>
        <START_TIME>2018-07-02T00:00:00.000</START_TIME>
        <END_TIME>2018-07-04T00:00:00.000</END_TIME>
        <definition name="zzz" version="1.0" />
        <source name="xxx" version="1.5" />
        <source name="yyy" version="a.b" />
      </metadata>
```

Figure 3-1: PRM Structure Example

```
      <!-- Pointing request data for the first request -->
      <!-- These is built either from templates in section 4 or -->
      <!-- constructed according to the steps in 5.4.4 -->
      <data>
      ...
      </data>

    &lt;/segment&gt;

    <!-- Second pointing request -->
    &lt;segment&gt;
      <metadata>
        <!--    Definitions for the second request -->
        <OBJECT_NAME>HIPPARCOS</OBJECT_NAME>
        <OBJECT_ID>090154</OBJECT_ID>
        <TIME_SYSTEM>UTC</TIME_SYSTEM>
        <START_TIME>2018-07-02T00:00:00.000</START_TIME>
        <END_TIME>2018-07-04T00:00:00.000</END_TIME>
        <source name="xxx" version="a.b" />
      </metadata>
      <data>
      ...
      </data>
    &lt;/segment&gt;
    .
    .
    .
    <!-- n-th pointing request -->
    &lt;segment&gt;
      <metadata>
        <OBJECT_NAME>HIPPARCOS</OBJECT_NAME>
        <OBJECT_ID>090154</OBJECT_ID>
        <TIME_SYSTEM>UTC</TIME_SYSTEM>
        <START_TIME>2018-07-02T00:00:00.000</START_TIME>
        <END_TIME>2018-07-04T00:00:00.000</END_TIME>
     <source name="xxx" version="1.5" />
      </metadata>
      <data>
      ...
      </data>
    &lt;/segment&gt;
  </body>

</prm>
```

Figure 3-1: PRM Structure Example (continued)

3.2.2 POINTING REQUEST DEFINITION ELEMENT STRUCTURE

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

3.2.2.2 The `<definition>` elements may contain the `name` attribute to allow its reference by the `<source>` element.

3.2.2.3 The `<definition>` elements may contain the `version` attribute to allow reference to different versions of the same definition by the `<source>` element.

3.2.2.4 If the definitions are incorporated in 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 Definitions shall specify elements used in the `<prm>` body, e.g., alignments, boresight directions, and directions to targets.

3.2.2.6 Definitions shall use the elements defined in table 3-1 and 3.3.2.

3.2.2.7 The definitions shall include exactly one *root frame* as the unique frame whose definition is not dependent on any other frame.

3.2.2.8 The definitions shall include one or more *secondary frames* defined relative to the *root frame* or to another *secondary frame* (see annex F).

3.2.3 POINTING REQUEST BODY ELEMENT STRUCTURE

3.2.3.1 The pointing request body shall describe the attitude of a spacecraft or any of its articulate parts over a period of time (attitude timeline).

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

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

3.2.3.4 The root element of the attitude timeline shall be the `<timeline>` element.

3.2.3.5 An attitude timeline shall consist of one or more attitude blocks defined by the `<block>` element.

3.2.3.6 Each `<block>` element shall define the attitude over a certain time interval.

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 3-1.

3.3.1.2 Depending on the specific use case, some of the physical or mathematical entity types defined in table 3-1 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 3.3.2.

3.3.1.4 All child elements and attributes which are not specified as mandatory 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 3-1: Overview of Entity Types Described by XML Elements

Entity type	Type description	Generic Element Name
Integer	Describes an integer number. An integer shall be dimensionless. Basic type is `xsd:integer`.	`<integer>`
List of integers	Describes a list of integers separated by white space. All integers in a list shall be dimensionless. List of integers may have any length.	`<integerList>`
Real	Describes a real number. The real can be dimensionless or have a unit (allowed units are listed in annex D). Basic type is `xsd:double`.	`<real>`
List of reals	Describes a list of reals separated by white space. All reals in a list have the same units. Allowed units are listed in annex D. List of reals may have any length.	`<reaList>`

Entity type	Type description	Generic Element Name
Epoch	Describes an instant in time. Epoch entities are used for instance to build timelines. Basic type is `ndm:epochType` (see reference [12]).	`<epoch>`
List of epochs	Describes a list of instants in time (epochs) separated by white space. List of times may have any length.	`<epochList>`
Duration	Describes an elapsed period of time. Duration entities are used to build epochs relative to other epochs. Basic type is `ndm:durationType` (see reference [12]).	`<duration>`
List of durations	Describes a list of elapsed times (durations) separated by white space. List of durations may have any length.	`<durationList>`
Direction vector	Describes a direction vector (unit vector, spherical coordinates 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.	`<dirVector>`
State Vector	Describes one orbital state defined as an epoch and the position and velocity at that epoch in Cartesian coordinates. Basic type is `ndm:stateVectorType` (see reference [13]).	`<stateVector>`
Orbit entity	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.	`<orbit>`
Surface	Describes a surface. A surface can be described in different ways depending on its type, e.g., a sphere is defined by its center and radius.	`<surface>`
Surface vector	Describes a trajectory over a surface.	`<surfaceVector>`
Reference frame entity	Describes a reference frame. Different types of reference frames can be defined (see annex F).	`<frame>`
Attitude block	Defines the attitude during a time interval.	`<block>`

Entity type	Type description	Generic Element Name
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.	`<attitude>`
Phase angle	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.	`<phaseAngle>`
Angular rate	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.	`<angularRate>`
Rotation	Defines rotation to be applied to a direction vector or attitude.	`<rotation>`
String	Contains string data.	`<string>`

NOTE – The tags provided in the Generic Element Name column indicates a default XML name for generic element use.

3.3.2 DETAILED DEFINITIONS OF POINTING REQUEST ELEMENTS

3.3.2.1 Epoch Type

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

Table 3-2: Epoch Type

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 3-1.)	<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>

[1] To ease reading, the notation '…' is used for elements whose representation is partial.

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.

Table 3-3: List of Epochs Type

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 3-1).	<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

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

Table 3-4: Duration Type

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

3.3.2.4 List of Durations Type

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

Table 3-5: List of Durations Type

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

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

Table 3-6: Integer Type

Representation	**Elements description**	**Example**
Integer value	Text contents of data type ***Integer***.	`<integer>1</integer>`

Integer operation	`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

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

Table 3-7: List of Integers Type

Representation	**Elements description**	**Example**

List of integers	Text contents of data type ***List of Integers.***	`<integerList>`
		`1 2 3`
		`</integerList>`

List of integers operation	`operator` attribute identifying the operation to be performed as well as the child elements over which the operation is performed.
	(See description of allowed list operators and child elements in annex C.)
		`<integerList operator='plus'>`
		`<integerList …>`
		`<integerList …>`
		`</integerList>`

3.3.2.7 Real Type

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

Table 3-8: Real Type

Representation	Elements description	Example
Real value	Text contents of data type ***Real***. Optional attribute `units` (see allowed values in annex D).	<real units="m">1.2</real>
Real operation	`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 C).	<real operator='plus'> <real>0.1</real> <real>0.2</real> </real>
Interpolation table	Child elements: `epochList` of type ***List of Epochs***, `valueList` of type ***List of Reals***. `derivativeList` of type ***List of Reals*** is optional. `interpolationAlgorithm` of type ***String***. `interpolationDegree` of type ***Integer***. 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 whose interpolation scheme is defined by `interpolationAlgorithm` and `interpolationDegree`. The `derivativeList` is optional for those algorithms requiring simultaneously the function values and their derivatives (e.g., splines).	<real> <epochList …> <valueList …> <derivativeList …> </real>

3.3.2.8 List of Reals Type

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

Table 3-9: List of Reals Type

Representation	Elements description	Example

List of reals	Text contents of data type ***List of Reals***.
	Optional `units` attribute (see annex D).
		`<realList>`
		` 1. 2. 3.`
		`</realList>`

List of reals 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 C.)
		`<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 shall be defined relative to a frame that can be fully referred to the root frame (see 3.3.2.14).

Table 3-10: Direction Vector Type

Representation	Elements description	Example
Coordinates	Optional attribute type (default value is cartesian). Allowed values for type: – cartesian (for which the content is a list of 3 real numbers representing a unit vector); – spherical (for which the content is a list of 2 real numbers representing the azimuth and polar angles); – raDec (for which the content is a list of 2 real numbers representing right ascension and declination). Mandatory 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 B. 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 D). 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>

Representation	Elements description	Example
Direction at epoch	dirVector element of ***Direction vector*** type plus refEpoch element of ***Epoch*** type. The resulting direction vector is the one 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. The child elements are of ***Direction vector*** type. The second child element is optional and will not be provided if operator value is derivative or unaryMinus. The frames of both direction vectors shall be defined relative to the same secondary frame or root frame (see 3.3.2.14).	<dirVector operator="cross"> <dirVector …> <dirVector …> </dirVector>
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

An orbital state shall be represented by an element of type ***State Vector*** with respect to a frame that can be fully referred to the root frame (see 3.3.2.14).

Table 3-11: State Vector Type

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 reference [6].
	Optional attribute `units` (see allowed values in annex D).
		`<stateVector>`
		`<epoch ...>`
		`<X ...>`
		`<Y ...>`
		`<Z ...>`
		`<XDOT ...>`
		`<YDOT ...>`
		`<ZDOT ...>`
		`</stateVector>`

3.3.2.11 Orbit Entity Type

The orbit entity type shall be used to describe the position of an object versus time with respect to a frame that can be fully referred to the root frame (see 3.3.2.14).

Table 3-12: Orbit Entity Type

Representation	Elements description	Example

Ephemerides object	`ephObject` element of data type ***String*** specifying the celestial object name contained in the ephemeris according to reference [9] as default.	`<orbit>`
		`<ephObject>`
		`MARS`
		`</ephObject>`
		`</orbit>`

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 according to reference [7].
		`<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 Type

The surface type shall be used to describe reference surfaces in a frame that can be fully referred to the root frame (see 3.3.2.14). All represented surfaces are differentiable and convex.

Table 3-13: Surface Type

Representation	Elements description	Example

Sphere	Mandatory 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	Mandatory 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 the semi axes of the ellipsoid (i.e., a, b, and c in the positive x, y, and z directions respectively) 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 Type

The surface vector type shall be used to describe reference trajectories over surfaces with respect to a frame that can be fully referred to the root frame (see 3.3.2.14).

Table 3-14: Surface Vector Type

Representation	Elements description	Example

Coordinates	`surface` element of type ***Surface***.
	`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.
		`<surfaceVector>`
		`<surface …>`
		`<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 shall not be aligned.
		`<surfaceVector>`
		`<surface …>`
		`<origin …>`
		`<targetDir …>`
		`</surfaceVector>`

3.3.2.14 Reference Frame Entity Type

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

3.3.2.14.2 None of the frames present in a PRM shall result undefined (this can be assured by being able to refer any frame to the root frame).

Table 3-15: Reference Frame Entity Type

Representation	Elements description	Example
Root frame	Mandatory `name` attribute of type string. Fixed value `baseFrame` attribute with value `none`. Only one root frame element is allowed. (See reference frames description in annex B.)	`<frame name='EME2000'` `baseFrame='none' />`
Secondary frame with undefined attitude	Mandatory `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 B and hierarchy in annex F.) The attitude of the frame may remain undefined until it is defined by a timeline element (see 3.3.3.4)	`<frame name='SC'` `baseFrame='EME2000' />`

Secondary frame with defined attitude	Mandatory `name` and `baseFrame` attributes of type string.
	Mandatory element `attitude` of type attitude.
	The `baseFrame` attribute shall correspond to the `name` of a previously defined `frame`.
	(See reference frames description in annex B and hierarchy in annex F.)
	The secondary frame `attitude` element is constructed by an Attitude Type element (see annex F)
		`<frame name='starTracker'`
		`baseFrame='SC'>`
		`<attitude …>`
		`<frame>`

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.

Table 3-16: Attitude Type

Representation	Elements description	Example

Directions	`frameDir` and `baseFrameDir` elements of type ***Direction vector***.
	`phaseAngle` element of ***Phase Angle*** type.
	(See attitude description in annex F.)
		`<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 F.)
		`<attitude>`
		`<attitude …>`
		`<rotation …>`
		`</attitude>`

3.3.2.16 Attitude Block Type

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

Table 3-17: Attitude Block Type

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 shall be defined relative to the respective frames of the parent attitude type element.

3.3.2.17.3 The directions corresponding to the `frameDir` and `baseFrameDir` elements shall be referred to fully defined attitudes.

3.3.2.17.4 For the directions in the phase angle 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.

Table 3-18: Phase Angle Type

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 phase angle element description in annex F.)
		`<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 phase angle element description in annex F.)
		`<phaseAngle>`
		`<frameDir …>`
		`<baseFrameDir …>`
		`<projAngle …>`
		`</phaseAngle>`

3.3.2.18 Angular Rate Type

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

Table 3-19: Angular Rate Type

Representation	Elements description	Example
Angular velocity	Optional attribute `units.` (See allowed values in annex D.)	`<angularRate units="deg/s">0.34 </angularRate>`

3.3.2.19 Rotation Type

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

Table 3-20: Rotation Type

Representation	Elements description	Example
Quaternion	quaternion of type ndm:quaternionType as defined in reference [13].	<rotation> <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' />
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 Type

The string type shall be used to describe string data.

Table 3-21: String Type

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 reference [5]).

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

Table 3-22: Include 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.

Table 3-23: Definition Element

Element description	Example

List of elements of any entity type as described in table 3-1 in any number.
The generic element name corresponding to the type (see table 3-1) shall be used.
	`<definition>`
	`  <real name='one'> 1. </real>`
	`  <real name='two'> 2. </real>`
	`</definition>`

3.3.3.3 Source Element

3.3.3.3.1 The `source` element shall be used to reference a `definition` element by its name.

Table 3-24: Source Element

Element description	Example

Reference to a definition element by its name.	`<definition name='myname'>`
	`  <real name='one'> 1. </real>`
	`  <real name='two'> 2. </real>`
	`</definition>`
	`<source name='myname' />`

3.3.3.4 Timeline Element

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

3.3.3.4.2 The `<timeline>` element shall always be a child of the PRM element.

3.3.3.4.3 The `<timeline>` element shall be composed of `<block>` elements sorted in chronological order.

3.3.3.4.4 Each `<block>` in the timeline shall provide the start of the block `<blockStart>`, the end of the block `<blockEnd>`, and the attitude definition `<attitude>`

Table 3-25: Timeline Element

Element description	Example

Sequence of one or more `block` elements of type ***Attitude block***.
The value of the `frame` attribute identifies one of the secondary fames previously defined.
	`<timeline frame='SC'>`
	`  <block>`
	`    <blockStart …>`
	`    <blockEnd …>`
	`    <attitude …>`
	`  <block …>`
	`  <block …>`
	`</timeline>`

3.4 THE NAMING AND REFERENCING MECHANISM

3.4.1 NAME ASSIGNMENT

3.4.1.1 Any element of the types defined in 3.3.2 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 3.3.2 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 Overview

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 which 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 Requirements

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

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

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

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

3.4.3.2.5 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.

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

3.4.3.2.7 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.

3.4.3.2.8 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).

3.4.3.2.9 The referencing parent element shall be built substituting each parameter in the referenced parent element with the corresponding child element in the referencing parent 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.

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

3.4.3.2.11 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 parent element then it is necessary to expand it in the referencing child element.

3.4.4 DISCUSSION—EXAMPLES

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

```
<dirVector name="axis1">
  <dirVector frame="EME2000" localName="Parameter1"> 0. 0. 1. </dirVector>
  <rotation>
    <!--- Naming of an element to be rotation1 --->
    <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 angle2 --->
      <!--- Parameter has default units but no default value --->
      <angle localName="angle2" units="deg" />
    </rotation>
  </rotation>
</dirVector>
```

Figure 3-2: Discussion Example Definition

Referencing and parameter substitution (conventional substitution):

```
<!--- Conventional substitution 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>
```

Figure 3-3: Discussion Example (Conventional Substitution)

Referencing and parameter substitution (with units overriding in substitution):

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

Figure 3-4: Discussion Example (with Units Overriding in Substitution)

Referencing and parameter substitution (all by default):

```
<!--- All substitution 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 must be present --->
<block ref="rotation2">
     <angle2>90.0</angle2>
</block>
```

Figure 3-5: Discussion Example (All by Default)

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 Templates provided in this section shall be combined in a single PRM following the rules in 3.2.

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 3.2 (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 GENERAL

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

a) An SC axis is pointed towards an inertial target.

b) The remaining degree of freedom in the SC attitude is determined by a phase angle from a reference inertial direction to another SC axis.

c) 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.

d) 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.

NOTE – The resulting SC attitude is defined in annex F.

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 F.

4.2.2 DEFINITION FILE TEMPLATE

4.2.2.1 Template 4-1 shall be used to build the definitions for a PRM containing inertial pointing requests.

NOTE – The variable content is shown by variable names between % symbols.

```xml
<definition name="%definitionName%" version="%definitionVersion%">
    <frame baseFrame="none" name="%baseFrameName%" />
    <frame baseFrame="%baseFrameName%" 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="%baseFrameName%"
                              coord="%phaseBaseCoordType%"
                              units="%phaseBaseFrameUnits%">%phaseBaseCoords%</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="%baseFrameName%"
                              coord="%offsetBaseCoordType%"
                              units="%offsetBaseFrameUnits%">%offsetBaseCoords%</baseFrameDir>
                <projAngle localName="offsetAngle" />
            </offsetAngle>
            <!-- Angular rate around the boresight -->
            <!-- Element optional; remove if no angular rate specified -->
            <angularRate localName="angularRate" />
        </attitude>
    </block>
</definition>
```

Template 4-1: Inertial Pointing Definition Template

4.2.2.2 The variable content in the definitions template shall be substituted according to the rules in table 4-1.

NOTE – The values provided in the Tag column are those in the container: `/prm/body/segment/metadata/definition/`.

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

Table 4-1: Inertial Pointing Definition File Variables

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`

`%baseFrameName%`	`frame[0]/@name`
	`frame[1]/@baseFrame`
	`block/attitude/phaseAngle/baseFrameDir/@frame`
	`block/attitude/offsetAngle/baseFrameDir/@frame`
		Base reference frame name.	One of the frames from annex B.	`EME2000`

`%spacecraftFrameName%`	`frame[1]/@name`
	`block/attitude/phaseAngle/frameDir/@frame`
	`block/attitude/offsetAngle/frameDir/@frame`
		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`
			`raDec`
				`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 shall 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 3-1	`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`
			`raDec`
				`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 shall 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 3-1	`0. 1. 0.`

Variable	**Tag**	**Description**	**Allowed values**	**Example value**

`%offsetBaseCoordType%`	`block/attitude/offsetAngle/baseFrameDir/@coord`	Type of coordinates defining the direction of the offset direction vector in inertial frame.	`cartesian`
			`spherical`
			`raDec`
				`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 shall 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 3-1	`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`
			`raDec`
				`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 shall be an empty string.	`deg`

Variable	Tag	Description	Allowed values	Example value
`%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 3-1	`0. 1. 0.`

4.2.3 REQUEST BODY TEMPLATE

4.2.3.1 Template 4-2 shall be used to build inertial pointing request blocks inside the PRM body.

NOTE – 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="%baseFrameName%"
                    coord="%inertialFrameCoordType%"
                    units="%inertialFrameUnits%">%inertialFrameCoords%</target>
            <!-- Phase angle, see convention in PRM annex F -->
            <!-- For spin stabilized spacecraft omit this block -->
            <phaseAngle units="%phaseAngleUnits%">%phaseAngle%</phaseAngle>
            <!-- Offset angle, see convention in PRM annex F -->
            <!-- Block optional; remove if no offset with respect to target -->
            <offsetAngle units="%offsetAngleUnits%">%offsetAngle%</offsetAngle>
            <!-- Angular rate around the boresight -->
            <!-- Element optional; remove if no angular rate specified -->
            <angularRate units="%angularRateUnits%">%angularRate%</angularRate>
        </block>
    </timeline>
</data>
```

Template 4-2: Inertial Pointing Request Block Template

4.2.3.2 The variable content in the pointing request block template shall be substituted according to the rules in table 4-2.

NOTE – The values provided in the Tag column are those in the container: `/prm/body/segment/data/timeline/block/`.

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 coordinates following the coordinates representation for direction vector type from 3.3.2.9.

Table 4-2: Inertial Pointing Request Block Variables

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 raDec	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 shall 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 3-1	0.052336 0. 0.99863
%baseFrameName%	target/@frame	Inertial reference frame name	One of the inertial frames from annex B	EME2000
%inertialFrameCoordType%	target/@coord	Type of the direction vector	cartesian spherical raDec	spherical

Variable	Tag	Description	Allowed values	Example value
`%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 shall 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 3-1	`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 3.3.2.7	`10.`
`%offsetAngleUnits%`	`offsetAngle/@units`	Units for the offset angle	`deg` `rad`	`deg`
`%offsetAngle%`	`offsetAngle`	The angular offset applied with respect to the reference direction.	Angle value according to the real value representation in 3.3.2.7	`10.`
`%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`

4.3 SUN POINTING

4.3.1 GENERAL

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

a) An SC axis is pointed towards the direction of the Sun (target).

b) 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 Template 4-3 shall be used to build the definitions for a PRM containing Sun pointing requests.

NOTE – The variable content is shown by variable names between % symbols.

```xml
<definition name="%definitionName%" version="%definitionVersion%">
    <frame baseFrame="none" name="%baseFrameName%" />
    <frame baseFrame="%baseFrameName%" 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>
            <!-- 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="%baseFrameName%"
                              coord="%phaseBaseCoordType%"
                              units="%phaseBaseFrameUnits%">%phaseBaseCoords%</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="%baseFrameName%"
                              coord="%offsetBaseCoordType%"
                              units="%offsetBaseFrameUnits%">%offsetBaseCoords%</baseFrameDir>
                <projAngle localName="offsetAngle" />
            </offsetAngle>
            <!-- Angular rate around the boresight -->
            <!-- Element optional; remove if no angular rate specified -->
            <angularRate localName="angularRate" />
        </attitude>
    </block>
</definition>
```

Template 4-3: Sun Pointing Definition Template

4.3.2.2 The variable content in the definitions template shall be substituted according to the rules in table 4-3.

NOTE – The values provided in the Tag column are those in the container: `/prm/body/segment/metadata/definition/`.

Table 4-3: Sun Pointing Definition File Variables

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`

`%baseFrameName%`	`frame[0]/@name`
	
	`frame[1]/@baseFrame`
	
	`frame[2]/@baseFrame`
	
	`block/attitude/phaseAngle/baseFrameDir/@frame`
	
	`block/attitude/offsetAngle/baseFrameDir/@frame`
		Base reference frame name.	One of the frames from annex B.	`EME2000`

`%spacecraftFrameName%`	`frame[1]/@name`
	
	`block/attitude/phaseAngle/frameDir/@frame`
	
	`block/attitude/offsetAngle/frameDir/@frame`
		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)	Valid URL	`/home/SC/ephem.oem`

`%phaseBaseCoordType%`	`block/attitude/phaseAngle/baseFrameDir/@coord`	Type of coordinates defining the direction of the phase direction vector in inertial frame.	`cartesian`
			`spherical`
			`raDec`
				`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 shall 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 3-1	`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`
			`raDec`
				`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 shall 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 3-1	`0. 1. 0.`

Variable	Tag	Description	Allowed values	Example value
%offsetBaseCoordType%	block/attitude/offsetAngle/baseFrameDir/@coord	Type of coordinates defining the direction of the offset direction vector in inertial frame.	cartesian spherical raDec	cartesian
%offsetBaseCoordType%	block/attitude/offsetAngle/baseFrameDir/@coord	Type of coordinates defining the direction of the offset direction vector in inertial frame.	cartesian spherical raDec	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 shall 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 3-1	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 raDec	cartesian

Variable	**Tag**	**Description**	**Allowed values**	**Example value**
`%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 shall 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 3-1	`0. 1. 0.`

4.3.3 REQUEST BODY TEMPLATE

4.3.3.1 Template 4-4 shall be used to build Sun pointing request blocks inside the PRM body.

NOTE – 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>
            <!-- Phase angle provides the rotation around the boresight -->
            <!-- For spin stabilized spacecraft omit this block -->
            <phaseAngle units="%phaseAngleUnits%">%phaseAngle%</phaseAngle>
            <!-- Offset angle, see convention in PRM annex F -->
            <!-- Block optional; remove if no offset with respect to target -->
            <offsetAngle units="%offsetAngleUnits%">%offsetAngle%</offsetAngle>
            <!-- Angular rate around the boresight -->
            <!-- Element optional; remove if no angular rate specified -->
            <angularRate units="%angularRateUnits%">%angularRate%</angularRate>
        </block>
    </timeline>
</data>
```

Template 4-4: Sun Pointing Request Block Template

4.3.3.2 The variable content in the pointing request block template shall be substituted according to the rules in table 4-4.

NOTE – The values provided in the Tag column are those in the container: `/prm/body/segment/data/timeline/block/`.

Table 4-4: Sun Pointing Request Block Variables

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`
			`raDec`
				`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 shall 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 3-1	`0.052336 0. 0.99863`
`%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 3.3.2.7	`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.`

Variable	Tag	Description	Allowed values	Example value
`%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.`

4.4 TRACK WITH INERTIAL DIRECTION YAW STEERING

4.4.1 GENERAL

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

a) An SC axis is pointed to a center of a solar system object (target).

b) The remaining degree of freedom in the SC attitude is determined by a phase angle from a reference inertial direction to a second SC axis.

c) The second SC axis and reference inertial direction used to define the phase are not parallel to the SC pointed axis and target direction respectively.

d) 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 F.

4.4.2 DEFINITION FILE TEMPLATE

4.4.2.1 Template 4-5 shall be used to build the definitions for a PRM containing track with inertial direction yaw steering requests.

NOTE – The variable content is shown by variable names between % symbols.

```
<definition name="%definitionName%" version="%definitionVersion%">
    <frame baseFrame="none" name="%baseFrameName%" />
    <frame baseFrame="%baseFrameName%" 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="%baseFrameName%"
                              coord="%phaseBaseCoordType%"
                              units="%phaseBaseFrameUnits%">%phaseBaseCoords%</baseFrameDir>
                <projAngle localName="phaseAngle" />
            </phaseAngle>
        </attitude>
    </block>
</definition>
```

Template 4-5: Track with Inertial Direction Yaw Steering Definition Template

4.4.2.2 The variable content in the definitions template shall be substituted according to the rules in table 4-5.

NOTE – The values provided in the Tag column are those in the container: `/prm/body/segment/metadata/definition/`.

Table 4-5: Track with Inertial Direction Yaw Steering Definition File Variables

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`

`%baseFrameName%`	`frame[0]/@name`
	`frame[1]/@baseFrame`
	`block/attitude/phaseAngle/baseFrameDir/@frame`
		Base reference frame name.	One of the frames from annex B.	`EME2000`

`%spacecraftFrameName%`	`frame[1]/@name` `block/attitude/phaseAngle/frameDir/@frame`	SC reference frame name	-	`SC`
`%spacecraftName%`	`orbit[0]/@name` `dirVector/origin/@ref`	SC name	-	`MEX`
`%OEM%`	`orbit[0]/orbitFile`	The URL to the orbit file containing the satellite trajectory (typically in OEM format)	Valid URL	

`%targetBodyName%`	`orbit[1]/@name`
	`orbit[1]/ephObject`
	`dirVector/target/@ref`
		The name of the body to be used as target for the pointing	Value given in reference [9]	`Mars`

`%targetOEM%`	`orbit[1]/orbitFile`	The URL to the orbit file containing the trajectory of the target object (typically in OEM format)	Valid URL	

Variable	Tag	Description	Allowed values	Example value
%phaseCoordType%	block/attitude/phaseAngle/frameDir/@coord	Type of coordinates defining the direction of the phase direction vector in SC frame.	cartesian spherical raDec	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 shall 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 3-1	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 raDec	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 shall be an empty string.	deg

Variable	Tag	Description	Allowed values	Example value
`%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 3-1	`0. 0. 1.`

4.4.3 REQUEST BODY TEMPLATE

4.4.3.1 Template 4-6 shall be used to build track with inertial direction yaw steering request blocks inside the PRM body.

NOTE – 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="%targetBodyName%" />
            <!-- Phase angle, see convention in annex F-->
            <phaseAngle units="%phaseAngleUnits%">%phaseAngle%</phaseAngle>
        </block>
    </timeline>
</data>
```

Template 4-6: Track with Inertial Direction Yaw Steering Request Block Template

4.4.3.2 The variable content in the pointing request block template shall be substituted according to the rules in table 4-6.

NOTE – The values provided in the Tag column are those in the container: `/prm/body/segment/data/timeline/block/`.

Table 4-6: Track with Inertial Direction Yaw Steering Pointing Request Block Variables

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`
			`raDec`
				`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 shall 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 3-1	`0.052336 0. 0.99863`
`%targetBodyName%`	`targetBody/@ref`	The name of the target body to be pointed	Value given in reference [9]	`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 GENERAL

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

a) An SC axis is pointed to a center of an object in the solar system (target).

b) A second SC axis is perpendicular to the Sun direction such that this second axis, the pointing direction and Sun direction are right handed.

NOTE – The Sun and the pointed SC axis direction are not parallel for any instant of time of the pointing request.

4.5.2 DEFINITION FILE TEMPLATE

4.5.2.1 Template 4-7 shall be used to build the definitions for a PRM containing track with power optimized yaw steering requests.

NOTE – The variable content is shown by variable names between % symbols.

```
<definition name="%definitionName%" version="%definitionVersion%">
    <frame baseFrame="none" name="%baseFrameName%" />
    <frame baseFrame="%baseFrameName%" 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 F-->
        <frameDir frame="%spacecraftFrameName%"
                  coord="%spacecraftCoordType%"
                  units="%spacecraftCoordUnits%">%spacecraftAxisPerpendicularToSun%</frameDir>
        <baseFrameDir ref="Sun" />
        <angle units="deg"> 90. </angle>
    </phaseAngle>
    <block name="bodyTrackWithPowerOptimizedYawSteering">
        <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>
```

Template 4-7: Track with Power Optimized Yaw Steering Definition Template

4.5.2.2 The variable content in the definitions template shall be substituted according to the rules in table 4-7.

NOTE – The values provided in the Tag column are those in the container: `/prm/body/segment/metadata/definition/`.

Table 4-7: Track with Power Optimized Yaw Steering Definition File Variables

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`
`%baseFrameName%`	`frame[0]/@name` `frame[1]/@baseFrame`	Base reference frame name.	One of the frames from annex B.	`EME2000`
`%spacecraftFrameName%`	`frame[1]/@name` `phaseAngle/frameDir/@frame`	SC reference frame name	-	`SC`

`%spacecraftName%`	`orbit[0]/@name`
	`dirVector[0]/origin/@ref`
	`dirVector[1]/origin/@ref`
		SC name	-	`MEX`

`%OEM%`	`orbit[0]/orbitFile`	The URL to the orbit file containing the satellite trajectory (typically in OEM format)	Valid URL	

`%targetBodyName%`	`orbit[1]/@name`
	`orbit[1]/ephObject`
	`dirVector[0]/target/@ref`
		The name of the body to be used as target for the pointing	Value given in reference [9]	`Mars`

`%targetOEM%`	`orbit[1]/orbitFile`	The URL to the orbit file containing the trajectory of the target object (typically in OEM format)	Valid URL	

Variable	Tag	Description	Allowed values	Example value

`%spacecraftCoordType%`	`phaseAngle/frameDir/@coord`	Coordinate type of the SC axis to be kept perpendicular to the Sun direction	`cartesian`
			`spherical`
			`raDec`
				`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 shall 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 Template 4-8 shall be used to build track with power optimized yaw steering request blocks inside the PRM body.

NOTE – The variable content is shown between % symbols.

```
<data>
    <timeline frame="%spacecraftFrameName%">
        <block ref="bodyTrackWithPowerOptimizedYawSteering">
            <!-- 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>
```

Template 4-8: Track with Power Optimized Yaw Steering Request Block Template

4.5.3.2 The variable content in the pointing request block template shall be substituted according to the rules in table 4-8.

NOTE – The values provided in the Tag column are those in the container: `/prm/body/segment/data/timeline/block/`.

Table 4-8: Track with Power Optimized Yaw Steering Pointing Request Block Variables

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`
			`raDec`
				`cartesian`

`%spacecraftCoordUnits%`	`boresight/@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 shall 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 3-1	`0.052336 0. 0.99863`

4.6 NADIR WITH POWER OPTIMIZED YAW STEERING

4.6.1 GENERAL

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

a) An SC axis is pointed such that the line along this axis intersects the surface of an object in nadir direction (target).

b) A second SC axis is perpendicular to the Sun direction such that this second axis, the pointing direction and Sun direction are right handed.

NOTE – The Sun and Nadir direction are not parallel for any instant of time of the pointing request.

4.6.2 DEFINITION FILE TEMPLATE

4.6.2.1 Template 4-9 shall be used to build the definitions for a PRM containing nadir pointing with power optimized yaw steering requests.

NOTE – The variable content is shown by variable names between % symbols.

```
<definition name="%definitionName%" version="%definitionVersion%">
    <frame baseFrame="none" name="%baseFrameName%" />
    <frame baseFrame="%baseFrameName%" name="%spacecraftFrameName%" />
    <orbit name="%spacecraftName%">
        <!-- OEM containing the SC orbit  -->
        <orbitFile>%OEM%</orbitFile>
    </orbit>
    <orbit name="Sun">
        <ephObject>SUN</ephObject>
    </orbit>
    <frame baseFrame="%baseFrameName%" name="%targetBodyName%">
        <attitude>
            <!-- Planet reference frame -->
            <rotation from="%baseFrameName%" 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 F-->
        <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>
```

Template 4-9: Nadir with Power Optimized Yaw Steering Definitions Template

4.6.2.2 The variable content in the definitions template shall be substituted according to the rules in table 4-9.

NOTE – The values provided in the Tag column are those in the container: `/prm/body/segment/metadata/definition/`.

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

Table 4-9: Nadir with Power Optimized Yaw Steering Definitions File Variables

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`

`%baseFrameName%`	`frame[0]/@name`
	`frame[1]/@baseFrame`
	`frame[2]/@baseFrame`
	`frame[2]/attitude/rotation/@from`
		Base reference frame name.	One of the frames from annex B.	`EME2000`

`%spacecraftFrameName%`	`frame[1]/@name` `phaseAngle/frameDir/@frame`	SC reference frame name	-	`SC`

`%spacecraftName%`	`orbit[0]/@name`
	`dirVector[0]/origin/@ref`
	`dirVector[1]/origin/@ref`
	`dirVector[1]/target/surfaceVector/origin/@ref`
		SC name	-	`MEX`

`%OEM%`	`orbit[0]/orbitFile`	The URL to the orbit file containing the satellite trajectory (typically in OEM format)	Valid URL	

Variable	Tag	Description	Allowed values	Example value
%targetBodyName%	frame[2]/@name orbit[2]/@name orbit[2]/ephObject surface/@frame surface/origin/@ref	The name of the body to be used as target for the pointing	Value given in reference [9]	Mars
%planetInertialFrame%	frame[2]/attitude/rotation/@to	Reference frame in the target body		IAUMars
%ellipsoidAxisUnits%	surface/a/@units surface/b/@units	Units for the dimension of the ellipsoid of the target body used to define the nadir pointing	km	km
%ellipsoidSemiMajorAxis%	surface/a	Size of the semimajor axis of the ellipsoid of the target body	-	6376.136
%ellipsoidSemiMinorAxis%	surface/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 raDec	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 shall be an empty string.	`deg`
`%spacecraftAxisPerpendicularToSun%`	`phaseAngle/frameDir`	Coordinates of the SC axis to be kept perpendicular to the Sun direction	-	`0. 0. 1.`

4.6.3 REQUEST BODY TEMPLATE

4.6.3.1 Template 4-10 shall be used to build the definitions for a PRM containing nadir pointing with power-optimized yaw steering requests.

NOTE – 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>
```

Template 4-10: Nadir with Power Optimized Yaw Steering Request Block Template

4.6.3.2 The variable content in the pointing request block template shall be substituted according to the rules in table 4-10.

NOTE – The values provided in the Tag column are those in the container: `/prm/body/segment/data/timeline/block/`.

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

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

Table 4-10: Nadir with Power Optimized Yaw Steering Pointing Request Block Variables

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`
			`raDec`
				`cartesian`

`%spacecraftCoordUnits%`	`boresight/@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 shall 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 3-1	`0.052336 0. 0.99863`

4.7 NADIR WITH GROUND TRACK ALIGNED YAW STEERING

4.7.1 GENERAL

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

a) An 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).

b) 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 form a right handed coordinate system.

c) The two SC axes are perpendicular to each other.

d) The ground track tangent in the surface fixed frame exists for any instant of time of the pointing request.

4.7.2 DEFINITION FILE TEMPLATE

4.7.2.1 Template 4-11 shall be used to build the definitions for a PRM containing nadir pointing with ground-track aligned yaw steering requests.

NOTE – The variable content is shown by variable names between % symbols.

```
<definition name="%definitionName%" version="%definitionVersion%">
    <frame baseFrame="none" name="%baseFrameName%" />
    <frame baseFrame="%baseFrameName%" name="%spacecraftFrameName%" />
    <orbit name="%spacecraftName%">
        <!-- OEM containing the SC orbit  -->
        <orbitFile>%OEM%</orbitFile>
    </orbit>
    <frame baseFrame="%baseFrameName%" name="%targetBodyName%">
        <attitude>
            <!-- Planet reference frame -->
            <rotation from="%baseFrameName%" 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 F-->
        <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>
```

Template 4-11: Nadir with Ground Track Aligned Yaw Steering Definitions Template

4.7.2.2 The variable content in the definitions template shall be substituted according to the rules in table 4-11.

NOTE – The values provided in the Tag column are those in the container: `/prm/body/segment/metadata/definition/`.

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

Table 4-11: Nadir with Ground Track Aligned Yaw Steering Definition File Variables

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`

`%baseFrameName%`	`frame[0]/@name`
	`frame[1]/@baseFrame`
	`frame[2]/@baseFrame`
	`frame[2]/attitude/rotation/@from`
		Base reference frame name.	One of the frames from annex B.	`EME2000`

`%spacecraftFrameName%`	`frame[1]/@name` `phaseAngle/frameDir/@frame`	SC reference frame name	-	`SC`

`%spacecraftName%`	`orbit[0]/@name`
	`dirVector[0]/origin/@ref`
	`dirVector[0]/target/surfaceVector/origin/@ref`
		SC name	-	`MEX`

`%OEM%`	`orbit[0]/orbitFile`	The URL to the orbit file containing the satellite trajectory (typically in OEM format)	Valid URL	

Variable	Tag	Description	Allowed values	Example value

`%targetBodyName%`	`frame[2]/@name`
	`orbit[1]/@name`
	`orbit[1]/ephObject`
	`surface/@frame`
	`surface/origin/@ref`
		The name of the body to be used as target for the pointing	Value given in reference [9]	`Mars`

`%planetInertialFrame%`	`frame[2]/attitude/rotation/@to`	Reference frame in the target body		`IAUMars`
`%ellipsoidAxisUnits%`	`surface/a/@units` `surface/b/@units`	Units for the dimension of the ellipsoid of the target body used to define the nadir pointing	km	`km`
`%ellipsoidSemiMajorAxis%`	`surface/a`	Size of the semimajor axis of the ellipsoid of the target body	-	`6376.136`
`%ellipsoidSemiMinorAxis%`	`surface/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`
			`raDec`
				`cartesian`

Variable	Tag	Description	Allowed values	Example value
`%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 shall be an empty string.	`deg`
`%spacecraftAxisPerpendicularToGroundTrack%`	`phaseAngle/frameDir`	Coordinates of the SC axis to be kept perpendicular to the ground track	-	`0. 0. 1.`

4.7.3 REQUEST BODY TEMPLATE

4.7.3.1 Template 4-12 shall be used to build nadir pointing with ground-track aligned yaw steering request blocks inside the PRM body.

NOTE – 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>
```

Template 4-12: Nadir with Ground Track Aligned Yaw Steering Request Block Template

4.7.3.2 The variable content in the pointing request block template shall be substituted according to the rules in table 4-12.

NOTE – The values provided in the Tag column are those in the container: `/prm/body/segment/data/timeline/block/`.

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 coordinates following the coordinates representation for direction vector type from 3.3.2.9.

Table 4-12: Nadir with Ground Track Aligned Yaw Steering Pointing Request Block Variables

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`
			`raDec`
				`cartesian`

`%spacecraftCoordUnits%`	`boresight/@units`	Units of the given pointed axis.	For `%spacecraftCoordType%=spherical`: `units="deg"` or `units="rad"`; for `%spacecraftCoordType%=cartesian` this variable shall 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 3-1	`0.052336 0. 0.99863`

4.8 NADIR WITH ORBITAL POLE ALIGNED YAW STEERING

4.8.1 GENERAL

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

a) An 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).

b) A second SC axis is aligned with the projection of the orbital pole (direction of the orbit angular momentum) in the plane perpendicular to the nadir direction.

4.8.2 DEFINITION FILE TEMPLATE

4.8.2.1 Template 4-13 shall be used to build the definitions for a PRM containing nadir pointing with orbital pole aligned yaw steering requests.

NOTE – The variable content is shown by variable names between % symbols.

```
<definition name="%definitionName%" version="%definitionVersion%">
    <frame baseFrame="none" name="%baseFrameName%" />
    <frame baseFrame="%baseFrameName%" name="%spacecraftFrameName%" />
    <orbit name="%spacecraftName%">
        <!-- OEM containing the SC orbit  -->
        <orbitFile>%OEM%</orbitFile>
    </orbit>
    <frame baseFrame="%baseFrameName%" name="%targetBodyName%">
        <attitude>
            <!-- Planet reference frame -->
            <rotation from="%baseFrameName%" 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 aligned with the orbit pole -->
        <!-- See signs convention in annex F -->
        <frameDir frame="%spacecraftFrameName%"
                  coord="%spacecraftCoordType%"
                  units="%spacecraftCoordUnits%">%spacecraftAxisParallelToOrbitPole%</frameDir>
        <baseFrameDir ref="orbitalPole"/>
        <angle 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>
```

Template 4-13: Nadir with Orbital Pole Aligned Yaw Steering Definitions Template

4.8.2.2 The variable content in the definitions template shall be substituted according to the rules in table 4-13.

NOTE – The values provided in the Tag column are those in the container: `/prm/body/segment/metadata/definition/`.

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

Table 4-13: Nadir with Orbital Pole Aligned Yaw Steering Definition File Variables

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
%baseFrameName%	frame[0]/@name frame[1]/@baseFrame frame[2]/@baseFrame frame[2]/attitude/rotation/@from	Base reference frame name.	One of the frames from annex B	EME2000
%spacecraftFrameName%	frame[1]/@name phaseAngle/frameDir/@frame	SC reference frame name	-	SC
%spacecraftName%	orbit[0]/@name dirVector[0]/origin/@ref dirVector[0]/target/surfaceVector/origin/@ref dirVector[1]/dirVector[0]/origin/@ref	SC name	-	MEX
%OEM%	orbit[0]/orbitFile	The URL to the orbit file containing the satellite trajectory (typically in OEM format)	Valid URL	

Variable	Tag	Description	Allowed values	Example value
%targetBodyName%	frame[2]/@name orbit[1]/@name orbit[1]/ephObject surface/@frame surface/origin/@ref dirVector[1]/dirVector[0]/target/@ref	The name of the body to be used as target for the pointing	Value given in reference [9]	Mars
%planetInertialFrame%	frame[2]/attitude/rotation/@to	Reference frame in the target body		IAUMars
%ellipsoidAxisUnits%	surface/a/@units surface/b/@units	Units for the dimension of the ellipsoid of the target body used to define the nadir pointing	km	km
%ellipsoidSemiMajorAxis%	surface/a	Size of the semimajor axis of the ellipsoid of the target body	-	6376.136
%ellipsoidSemiMinorAxis%	surface/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 raDec	cartesian

Variable	Tag	Description	Allowed values	Example value
%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 shall be an empty string.	deg
%spacecraftAxisParallelToOrbitPole%	phaseAngle/frameDir	Coordinates of the SC axis to be kept parallel to the orbit pole	-	0. 0. 1.

4.8.3 REQUEST BODY TEMPLATE

4.8.3.1 Template 4-14 shall be used to build nadir pointing with orbital pole aligned yaw steering request blocks inside the PRM body.

NOTE – 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>
```

Template 4-14: Nadir with Orbital Pole Aligned Yaw Steering Request Block Template

4.8.3.2 The variable content in the pointing request block template shall be substituted according to the rules in table 4-14.

NOTE – The values provided in the Tag column are those in the container: `/prm/body/segment/data/timeline/block/`.

Table 4-14: Nadir with Orbital Pole Aligned Yaw Steering Pointing Request Block Variables

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`
			`raDec`
				`cartesian`

`%spacecraftCoordUnits%`	`boresight/@units`	Units of the given pointed axis	For `%spacecraftCoordType%=spherical`: `units="deg"` or `units="rad"`; for `%spacecraftCoordType%=cartesian` this variable shall 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 3-1	`0.052336 0. 0.99863`

4.9 LIMB POINTING WITH POWER OPTIMIZED YAW STEERING

4.9.1 GENERAL

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

a) An SC axis is pointed towards a point (target) that lies at a specified height along the local normal over a point on the limb of an object.

b) The point on the limb is defined as the intersection of the limb with the half-plane defined by the SC to object-center direction and a reference inertial direction.

c) The reference inertial direction shall not be aligned with the SC to object-center direction.

d) A second SC axis shall point in a direction perpendicular to the Sun direction such that this second axis, the pointing direction and Sun direction are right handed.

e) The two SC axes shall be perpendicular to each other.

4.9.2 DEFINITION FILE TEMPLATE

4.9.2.1 Template 4-15 shall be used to build the definitions for a PRM containing limb pointing with power optimized yaw steering requests.

NOTE – The variable content is shown by variable names between % symbols.

```
<definition name="%definitionName%" version="%definitionVersion%">
    <frame baseFrame="none" name="%baseFrameName%" />
    <frame baseFrame="%baseFrameName%" 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="%baseFrameName%" name="%targetBodyName%">
        <attitude>
            <!-- Planet reference frame -->
            <rotation from="%baseFrameName%" 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 F-->
        <frameDir frame="%spacecraftFrameName%"
                  coord="%spacecraftCoordType%"
                  units="%spacecraftCoordUnits%">%spacecraftAxisPerpendicularToSun%</frameDir>
        <baseFrameDir ref="Sun"/>
        <angle units="deg"> 90. </angle>
    </phaseAngle>
    <block name="limbWithPowerOptimizedYawSteering">
        <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>
```

Template 4-15: Limb Pointing with Power Optimized Yaw Steering Definitions Template

4.9.2.2 The variable content in the definitions template shall be substituted according to the rules in table 4-15.

NOTE – The values provided in the Tag column are those in the container: `/prm/body/segment/metadata/definition/`.

Table 4-15: Limb Pointing with Power Optimized Yaw Steering Definition File Variables

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
%baseFrameName%	frame[0]/@name frame[1]/@baseFrame frame[2]/@baseFrame frame[2]/attitude/rotation/@from	Base reference frame name.	One of the frames from annex B.	EME2000
%spacecraftFrameName%	frame[1]/@name phaseAngle/frameDir/@frame	SC reference frame name	-	SC
%spacecraftName%	orbit[0]/@name dirVector/origin/@ref	SC name	-	MEX
%OEM%	orbit[0]/orbitFile	The URL to the orbit file containing the satellite trajectory (typically in OEM format)	Valid URL	
%targetBodyName%	orbit[1]/@name orbit[1]/ephObject frame[2]/@name surface/@frame surface/origin/@ref	The name of the body to be used as target for the pointing	Value given in reference [9]	Mars
%planetInertialFrame%	frame[2]/attitude/rotation/@to	Reference frame in the target body		IAUMars

Variable	Tag	Description	Allowed values	Example value

`%ellipsoidAxisUnits%`	`surface/a/@units`
	
	`surface/b/@units`
		Units for the dimension of the ellipsoid of the target body used to define the limb pointing	`km`	`km`

`%ellipsoidSemiMajorAxis%`	`surface/a`	Size of the semimajor axis of the ellipsoid of the target body	-	`6376.136`
`%ellipsoidSemiMinorAxis%`	`surface/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`
			`raDec`
				`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 shall 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 Template 4-16 shall be used to build limb pointing with power optimized yaw steering request blocks inside the PRM body.

NOTE – The variable content is shown between % symbols.

```
<data>
    <timeline frame="%spacecraftFrameName%">
        <block ref="limbWithPowerOptimizedYawSteering">
            <!-- 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%">%limbCoords%</dirVector>
                <height units ="%heightUnits%">%height%</height>
            </target>
        </block>
    </timeline>
</data>
```

Template 4-16: Limb Pointing with Power Optimized Yaw Steering Request Block Template

4.9.3.2 The variable content in the pointing request block template shall be substituted according to the rules in table 4-16.

NOTE – The values provided in the Tag column are those in the container: `/prm/body/segment/data/timeline/block/`.

Table 4-16: Limb Pointing with Power Optimized Yaw Steering Pointing Request Block Variables

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 raDec	cartesian
%spacecraftCoordUnits%	boresight/@units	Units of the given pointed axis	For %spacecraftCoordType%=spherical: units="deg" or units="rad"; for %phaseBaseCoordType%=cartesian this variable shall 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 3-1	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 raDec	cartesian

Variable	Tag	Description	Allowed values	Example value
`%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 shall be an empty string.	`deg`
`%limbCoords%`	`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 GENERAL

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

a) An SC axis is pointed towards a point (target) that lies at a specified height along the local normal over a point on the limb of an object.

b) The point on the limb is defined as the intersection of the limb with the half-plane defined by the SC to object-center direction and a reference inertial direction.

c) The reference inertial direction shall not be aligned with the SC to object-center direction.

d) The remaining degree of freedom in the SC attitude is determined by a phase angle from the reference inertial direction to another SC axis.

e) 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.

f) 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 F.

4.10.2 DEFINITION FILE TEMPLATE

4.10.2.1 Template 4-17 shall be used to build the definitions for a PRM containing limb pointing with inertial direction yaw steering requests.

NOTE – The variable content is shown by variable names between % symbols.

```
<definition name="%definitionName%" version="%definitionVersion%">
    <frame baseFrame="none" name="%baseFrameName%" />
    <frame baseFrame="%baseFrameName%" 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="%baseFrameName%" name="%targetBodyName%">
        <attitude>
            <!-- Planet reference frame -->
            <rotation from="%baseFrameName%" 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="%baseFrameName%"
                      coord="%phaseBaseCoordType%"
                      units="%phaseBaseFrameUnits%">%phaseBaseCoords%</baseFrameDir>
        <projAngle localName="phaseAngle" />
    </phaseAngle>
    <block name="limbWithInertialPointingYawSteering">
        <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>
```

Template 4-17: Limb Pointing with Inertial Direction Yaw Steering Definitions Template

4.10.2.2 The variable content in the definitions template shall be substituted according to the rules in table 4-17.

NOTE – The values provided in the Tag column are those in the container: `/prm/body/segment/metadata/definition/`.

Table 4-17: Limb Pointing with Inertial Direction Yaw Steering Definition File Variables

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
%baseFrameName%	frame[0]/@name frame[1]/@baseFrame frame[2]/@baseFrame frame[2]/attitude/rotation/@from phaseAngle/baseFrameDir/@frame	Base reference frame name.	One of the frames from annex B.	EME2000
%spacecraftFrameName%	frame[1]/@name phaseAngle/frameDir/@frame	SC reference frame name	-	SC
%spacecraftName%	orbit[0]/@name	SC name	-	MEX
%OEM%	orbit[0]/orbitFile	The URL to the orbit file containing the satellite trajectory (typically in OEM format)	Valid URL	
%targetBodyName%	orbit[1]/@name orbit[1]/ephObject frame[2]/@name surface/@frame surface/origin/@ref	The name of the body to be used as target for the pointing	Value given in reference [9]	Mars

Variable	Tag	Description	Allowed values	Example value
`%planetInertialFrame%`	`frame[2]/attitude/rotation/@to`	Reference frame in the target body		`IAUMars`
`%ellipsoidAxisUnits%`	`surface/a/@units` `surface/b/@units`	Units for the dimension of the ellipsoid of the target body used to define the nadir pointing	`km`	`km`
`%ellipsoidSemiMajorAxis%`	`surface/a`	Size of the semimajor axis of the ellipsoid of the target body	-	`6376.136`
`%ellipsoidSemiMinorAxis%`	`surface/b`	Size of the semiminor axis of the ellipsoid of the target body	-	`6256.345`

`%phaseCoordType%`	`phaseAngle/frameDir/@coord`	Type of coordinates defining the direction of the phase direction vector in SC frame.	`cartesian`
			`spherical`
			`raDec`
				`cartesian`

`%phaseFrameUnits%`	`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 shall be an empty string.	`deg`
`%phaseCoords%`	`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 3-1	`0. 1. 0.`

Variable	**Tag**	**Description**	**Allowed values**	**Example value**

`%phaseBaseCoordType%`	`phaseAngle/baseFrameDir/@coord`	Type of coordinates defining the direction of the phase direction vector in inertial frame.	`cartesian`
			`spherical`
			`raDec`
				`cartesian`

`%phaseBaseFrameUnits%`	`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 shall be an empty string.	`deg`
`%phaseBaseCoords%`	`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 3-1	`0. 0. 1.`

4.10.3 REQUEST BODY TEMPLATE

4.10.3.1 Template 4-18 shall be used to build limb pointing with inertial direction yaw steering request blocks inside the PRM body.

NOTE – The variable content is shown between % symbols.

```
<data>
    <timeline frame="%spacecraftFrameName%">
        <block ref="limbWithInertialPointingYawSteering">
            <!-- 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%">%limbCoords%</dirVector>
                <height units ="%heightUnits%">%height%</height>
            </target>
            <!-- Phase angle, see convention in annex F-->
            <phaseAngle units="%phaseAngleUnits%">%phaseAngle%</phaseAngle>
        </block>
    </timeline>
</data>
```

Template 4-18: Limb Pointing with Inertial Direction Yaw Steering Request Block Template

4.10.3.2 The variable content in the pointing request block template shall be substituted according to the rules in table 4-18.

NOTE – The values provided in the Tag column are those in the container: `/prm/body/segment/data/timeline/block/`.

Table 4-18: Limb Pointing with Inertial Direction Yaw Steering Pointing Request Block Variables

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`
			`raDec`
				`cartesian`

`%spacecraftCoordUnits%`	`boresight/@units`	Units of the given pointed axis	For %spacecraftCoordType%=s`pherical`: `units="deg"` or `units="rad"`; for `%phaseBaseCoordType%=cartesian` this variable shall 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 3-1	`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`
			`raDec`
				`cartesian`

Variable	Tag	Description	Allowed values	Example value
%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 shall be an empty string.	deg
%limbCoords%	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
%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 GENERAL

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

a) An SC axis is pointed towards the SC velocity relative to another object.

b) The remaining degree of freedom in the SC attitude is determined by a phase angle from the SC orbital pole (direction of the angular momentum) with respect to the object and another SC axis.

c) The two SC axes shall not be parallel.

d) 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 second SC axis, a positive angle meaning a positive rotation around the target direction. The resulting SC attitude is defined in annex F.

4.11.2 DEFINITION FILE TEMPLATE

4.11.2.1 Template 4-19 shall be used to build the definitions for a PRM containing velocity pointing with orbital pole yaw steering requests.

NOTE – The variable content is shown by variable names between % symbols.

```xml
<definition name="%definitionName%" version="%definitionVersion%">
    <frame baseFrame="none" name="%baseFrameName%" />
    <frame baseFrame="%baseFrameName%" 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>
```

Template 4-19: Velocity Pointing with Orbital Pole Yaw Steering Definitions Template

4.11.2.2 The variable content in the definitions template shall be substituted according to the rules in table 4-19.

NOTE – The values provided in the Tag column are those in the container: `/prm/body/segment/metadata/definition/`.

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

Table 4-19: Velocity Pointing with Orbital Pole Yaw Steering Definition File Variables

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`
`%baseFrameName%`	`frame[0]/@name` `frame[1]/@baseFrame`	Base reference frame name.	One of the frames from annex B.	`EME2000`
`%spacecraftFrameName%`	`frame[1]/@name` `block/attitude/phaseAngle/frameDir/@frame`	SC reference frame name	-	`SC`

`%spacecraftName%`	`orbit[0]/@name`
	`dirVector[0]/target/@ref`
	`dirVector[1]/target/@ref`
		SC name	-	`MEX`

`%OEM%`	`orbit[0]/orbitFile`	The URL to the orbit file containing the satellite trajectory (typically in OEM format)	Valid URL	

`%targetBodyName%`	`orbit[1]/@name`
	`orbit[1]/ephObject`
	`dirVector[0]/origin/@ref`
	`dirVector[1]/origin/@ref`
		The name of the body to be used as target for the pointing	Value given in reference [9]	`Mars`

`%phaseCoordType%`	`block/attitude/phaseAngle/frameDir/@coord`	Type of coordinates defining the direction of the phase direction vector in SC frame.	`cartesian`
			`spherical`
			`raDec`
				`cartesian`

Variable	Tag	Description	Allowed values	Example value
`%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 shall 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 3-1	`0. 1. 0.`

4.11.3 REQUEST BODY TEMPLATE

4.11.3.1 Template 4-20 shall be used to build velocity pointing with orbital pole yaw steering request blocks inside the PRM body.

NOTE – 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>
```

Template 4-20: Velocity Pointing with Orbital Pole Yaw Steering Request Block Template

4.11.3.2 The variable content in the pointing request block template shall be substituted according to the rules in table 4-20.

NOTE – The values provided in the Tag column are those in the container: `/prm/body/segment/data/timeline/block/`.

Table 4-20: Velocity Pointing with Orbital Pole Yaw Steering Pointing Request Block Variables

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 raDec	cartesian
%spacecraftCoordUnits%	boresight/@units	Units of the SC main pointing axis	For %spacecraftCoordType% =spherical: units="deg" or units="rad"; for %spacecraftCoordType% =cartesian this variable shall 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 3-1	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

5.1 OVERVIEW

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. The process described in this section was used to create the templates described in section 4.

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 (there are two ways in which the remaining degree of freedom can be resolved; see annex F item 2)) 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).

5.2 GENERAL RULES

NOTE – There are two essential elements in the construction of a PRM.

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

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

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

5.3 PRM HIGH LEVEL STRUCTURE

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

5.3.2 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>
```

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

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

5.3.5 The user shall provide the value for the `CREATION_DATE` element consistently with the date of creation of the PRM in UTC (see reference [1], ASCII Time Code A or B).

5.3.6 The user shall provide the value for the `ORIGINATOR` element following the rules in annex E.

NOTE – The detailed structure of the `body` section depends on the actual nature of the PRM being built; details are provided in the following subsections.

5.4 PRM SEGMENT

5.4.1 OVERVIEW

This subsection focuses in the actual pointing request aspects of the PRM. The PRM `body` contains a number of `segment` elements providing the details of the pointing request. As described in 3.2, the two main constituents of any PRM are the `definition` element in the `metadata` container and the request `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 3.4.2 and 3.4.3.

5.4.2 REFERENCE STRUCTURE

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

```
&lt;segment&gt;
    <metadata>
        <OBJECT_NAME>HIPPARCOS</OBJECT_NAME>
        <OBJECT_ID>090154</OBJECT_ID>
        <TIME_SYSTEM>UTC</TIME_SYSTEM>
        <START_TIME>2018-07-01T00:00:00.000</START_TIME>
        <STOP_TIME>2018-07-03T00:00:00.000</STOP_TIME>
        <definition>
        </definition>
        ...
    </metadata>
    <data>
        <timeline>
            <block>
            </block>
            ...
        </timeline>
        ...
    </data>
&lt;/segment&gt;
```

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

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

5.4.3 DEFINITION SECTION

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

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

5.4.3.3 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>
```

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

5.4.3.5 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, i.e., from reference [9].

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

NOTES

1 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>
```

2 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.

5.4.3.6.1 The base frame shall be defined in the definition section.

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

5.4.3.6.2 Every reference frame in the definition 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>
```

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

5.4.3.7.1 The definition 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>
```

5.4.3.7.2 The definition section shall identify all required celestial bodies trajectories through their common designators (according to reference [9]).

```
<definition name="defBlock" version="a.b">
    <frame baseFrame="none" name="EME2000" />
    <frame baseFrame="EME2000" name="SC" />
    <orbit name="SC#1">
        <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 reference [9] is the value of the element `ephObject`.

5.4.3.8 The definition section shall define all reference directions needed to define the request, based on the frames and objects in the definition section.

NOTE – 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>
```

NOTE – 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.

5.4.3.9 The definition 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>
```

5.4.3.9.1 Each attitude block with the definition section shall provide block start and block end identifiers.

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

5.4.3.9.3 The reason not to give actual epochs but unique identifiers 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>
```

5.4.3.10 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>
```

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

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

5.4.3.10.3 The definition of the **direction to be pointed** shall use the reference to the directions resulting from the implementation of steps 5.4.3.10.1 and 5.4.3.10.2.

5.4.3.10.4 The definition of the **direction to point to** shall use the reference to the directions resulting from the implementation of steps 5.4.3.10.1 and 5.4.3.10.2.

```
<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>
```

NOTE – 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. It should be noted that the reference directions had already being defined in step 5.4.3.8, and 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`).

5.4.3.11 The definition section shall close any remaining degree of freedom.

5.4.3.12 The attitude section may contain the definition of a phase around the spacecraft pointing direction (e.g., boresight), in which 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>
```

NOTE – 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.

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

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

5.4.3.12.3 The plane normal may be defined in different ways:

a) with a fixed unit vector in inertial space, e.g.,

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

b) with the orbit pole, e.g.,

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

NOTE – 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.

5.4.3.12.4 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>
```

NOTE – 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).

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

5.4.3.14 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>
```

5.4.3.15 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>
```

5.4.4 REQUEST SECTION

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

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

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

NOTES

1 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.

2 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>
```

5.4.4.4 Request Section Step by Step (Reference Case)

5.4.4.4.1 The pointing request associated to the PRM definition shall define at least one block in the timeline.

5.4.4.4.2 Each block in the timeline shall provide its start and end time and the attitude definition.

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

5.4.4.4.3 The pointing request associated to the PRM definition may define several blocks in the timeline in chronological order.

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

5.4.4.5 Request Section Step by Step (Configurable Boresight)

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

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

NOTE – 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:

```
<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>
```

5.4.4.5.3 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>
```

NOTE – 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.

5.4.4.6 Request Section Step by Step (Configurable Target)

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

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

NOTE – 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:

```
    <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>
```

5.4.4.6.3 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>
```

NOTE – 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

IMPLEMENTATION CONFORMANCE STATEMENT PROFORMA

(NORMATIVE)

A1 INTRODUCTION

A1.1 OVERVIEW

This annex provides the Implementation Conformance Statement (ICS) Requirements List (RL) for an implementation of Pointing Request Message (CCSDS 509.0). The ICS for an implementation is generated by completing the RL in accordance with the instructions below. An implementation shall satisfy the mandatory conformance requirements referenced in the RL.

The RL in this annex is blank. An implementation's completed RL is called the ICS. The ICS states which capabilities and options have been implemented. The following can use the ICS:

– the implementer, as a checklist to reduce the risk of failure to conform to the standard through oversight;

– a supplier or potential acquirer of the implementation, as a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard ICS proforma;

– a user or potential user of the implementation, as a basis for initially checking the possibility of interworking with another implementation (it should be noted that, while interworking can never be guaranteed, failure to interwork can often be predicted from incompatible ICSes);

– a tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

A1.2 ABBREVIATIONS AND CONVENTIONS

The RL consists of information in tabular form. The status of features is indicated using the abbreviations and conventions described below.

<u>Item Column</u>

The item column contains sequential numbers for items in the table.

Feature Column

The feature column contains a brief descriptive name for a feature. It implicitly means 'Is this feature supported by the implementation?'

NOTE – The features itemized in the RL are elements of a PRM. Therefore support for a mandatory feature indicates that generated messages will include that feature, and support for an optional feature indicates that generated messages can include that feature.

Keyword Column

The keyword column contains, where applicable, the PRM keyword associated with the feature.

Reference Column

The reference column indicates the relevant subsection or table in *Pointing Request Message* (CCSDS 509.0) (this document).

Status Column

The status column uses the following notations:

M mandatory.

O optional.

Support Column Symbols

The support column is to be used by the implementer to state whether a feature is supported by entering Y, N, or N/A, indicating:

Y Yes, supported by the implementation.

N No, not supported by the implementation.

N/A Not applicable.

A1.3 INSTRUCTIONS FOR COMPLETING THE RL

An implementer shows the extent of compliance to the Recommended Standard by completing the RL; that is, the state of compliance with all mandatory requirements and the options supported are shown. The resulting completed RL is called an ICS. The implementer shall complete the RL by entering appropriate responses in the support or values supported column, using the notation described in A1.2. If a conditional requirement is inapplicable, N/A should be used. If a mandatory requirement is not satisfied, exception information must be supplied by entering a reference Xi, where i is a unique identifier, to an accompanying rationale for the noncompliance.

A2 ICS PROFORMA FOR POINTING REQUEST MESSAGE

A2.1 GENERAL INFORMATION

A2.1.1 Identification of ICS

Date of Statement (DD/MM/YYYY)	
ICS serial number	
System Conformance statement cross-reference	

A2.1.2 Identification of Implementation Under Test (IUT)

Implementation name	
Implementation version	
Special Configuration	
Other Information	

A2.1.3 Identification of Supplier

Supplier	
Contact Point for Queries	
Implementation Name(s) and Versions	
Other information necessary for full identification, e.g., name(s) and version(s) for machines and/or operating systems; System Name(s)	

A2.1.4 Document Version

CCSDS 509.0 Document Version	
Have any exceptions been required? NOTE – A YES answer means that the implementation does not conform to the Recommended Standard. Non-supported mandatory capabilities are to be identified in the ICS, with an explanation of why the implementation is non-conforming.	Yes _____ No______

A2.1.5 Requirements List

Item	Feature	Keyword	Reference	Status	Support
1	PRM Header	`header`	5.3.2	M	
2	PRM version	`CCSDS_PRM_VERS`	5.3.2	M	
4	Message creation date/time	`CREATION_DATE`	5.3.2	M	
5	Message originator	`ORIGINATOR`	5.3.2	M	
6	PRM body	`body`	5.3.2	M	
7	PRM Segment	`segment`	5.3.2	M	
8	PRM metadata block	`metadata`	5.3.2	M	
9	Satellite name	`OBJECT_NAME`	5.4.2.1	M	
10	Satellite identifier	`OBJECT_ID`	5.4.2.1	M	
11	Time system	`TIME_SYSTEM`	5.4.2.1	M	
12	Aggregated timelines start time	`START_TIME`	5.4.2.1	M	
13	Aggregated timelines stop time	`STOP_TIME`	5.4.2.1	M	
14	Definition block	`definition`	5.4.3.2	O	
15	Root frame	`frame`	5.4.3.6.1	M	
16	Secondary frame	`frame`	5.4.3.6.2	M	
17	Spacecraft trajectory	`orbit`	5.4.3.7.1	M	
18	Celestial body trajectory	`orbit`	5.4.3.7.2	O	
19	Pointing direction	`dirVector`	5.4.3.8	M	
20	Attitude data block (defining)	`block`	5.4.3.9	M	
21	Attitude block start time	`startEpoch`	5.4.3.9.3	M	
22	Attitude block end time	`endEpoch`	5.4.3.9.3	M	
23	Attitude definition	`attitude`	5.4.3.10	M	
24	Reference to another definition block	`source`	Figure 3-1	O	
25	PRM data block	`data`	5.3.2	O	
26	Attitude timeline	`timeline`	5.4.4.4.1	M	
27	Attitude block (reference or defining)	`block`	5.4.4.4.1	M	
28	Attitude block start time	`blockStart`	5.4.4.4.2	M	
29	Attitude block end time	`blockEnd`	5.4.4.4.2	M	
30	Attitude (reference or defining)	`attitude`	5.4.4.4.2	M	

ANNEX B

VALUES FOR TIME_SYSTEM AND FRAME RELATED TAGS

(NORMATIVE)

The values in this annex represent the set of acceptable values for the <TIME_SYSTEM> element and the <frameEntity> element (attribute name) in the PRM. If exchange partners wish to use different settings, the settings should be documented in the ICD.

B1 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
TCG	Geocentric Coordinate Time
TDB	Barycentric Dynamical 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 MET, or of the event for MRT, 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> element, 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. The time format should utilize only three-digit days from the MET or MRT epoch, not months and days of the months.

B2 REFERENCE FRAME

Reference Frame Value	Meaning
EME2000	Earth Mean Equator and Equinox of J2000
GCRF	Geocentric Celestial Reference Frame
GRC	Greenwich Rotating Coordinates
ICRFYYYY	International Celestial Reference Frame of year YYYY
InstrX	Placeholder for any instrument reference frame
ITRFYYYY	International Terrestrial Reference Frame of year YYYY
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
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.
TOD	True of Date

ANNEX C

LIST OPERATORS

(NORMATIVE)

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 shall 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 shall 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 shall 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*** shall 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 shall 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
<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) `accumulate`: 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='accumulate'> <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 shall have the same lengths and unit type. The fist list contains the list of values to be derived; the second list contains the independent variable to be used in 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 D

SUPPORTED UNITS

(NORMATIVE)

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
		`as`	Seconds of arc
Angular velocity	`deg/s`	`deg/s`	Degrees per second
		`deg/min`	Degrees per minute
		`rad/s`	Radians per second
		`as/s`	Second of arc per second
Distance	`km`	`Km`	Kilometers
		`m`	Meters
Duration	`s`	`s`	Seconds
		`min`	Minutes
		`h`	Hours
		`d`	Days
		`dhms`	Duration specified in calendar format ([+-][[[dddT]hh:]mm:]ss[.ss])

ANNEX E

SECURITY, SANA, AND PATENT CONSIDERATIONS

(INFORMATIVE)

E1 SECURITY CONSIDERATIONS

E1.1 ANALYSIS OF SECURITY CONSIDERATIONS

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

E1.2 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.

E1.3 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, (b) unauthorized access to the messages during transmission between exchange partners and (c) modification of the messages between 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.

E1.4 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.

E1.5 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.

E1.6 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.

E1.7 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.

E1.8 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.

E1.9 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.

E1.10 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.

E1.11 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.

E2 SANA CONSIDERATIONS

The following PRM related items are registered with the SANA Operator.

– The PRM XML templates (see reference [14]).[2]

The following PRM elements should be from the SANA registry:

– the spacecraft names that appear as origin and target in the PRM (see reference [10]);

– the PRM originators (see reference [11]).

The use of reference [9] is a convenient solution of the identification of celestial bodies in absence of a corresponding SANA reference. For spacecraft the common identifiers in the SANA registry shall be preferred.

The registration rule for new entries in the registry is the approval of new requests by the CCSDS Area or Working Group responsible for the maintenance of the PRM at the time of the request. New requests for this registry should be sent to SANA (mailto:info@sanaregistry.org).

E3 PATENT CONSIDERATIONS

The recommendations of this document have no patent issues.

[2] The PRM templates downloaded from the SANA registry are XML compliant in structure and coding. The download process should ensure that these are not modified. Also, the editing should preserve structure and encoding; i.e., a proper XML editor or a plain text editor must be used; a word processing application like MS-Word must not be used.

ANNEX F

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 secondary 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 secondary frame is defined as follows.

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

$$M = \begin{pmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1 q_2 + q_3 q_4) & 2(q_1 q_3 - q_2 q_4) \\ 2(q_1 q_2 - q_3 q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2 q_3 + q_1 q_4) \\ 2(q_1 q_3 + q_2 q_4) & 2(q_2 q_3 - q_1 q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{pmatrix}$$

   such that

$$v_{\text{secondary frame}} = M \cdot v_{\text{base frame}}$$

   being $v_{\text{secondary frame}}$ and $v_{\text{base frame}}$ column vectors.

   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 phase angle (see XML snippets below).

   – Two vectors relative to the secondary frame are given by the `frameDir` elements.

   – Two vectors relative to the base frame are given by the `baseFrameDir` elements.

   – The secondary frame attitude results from aligning the direction vector defined in the secondary frame (`attitude/frameDir`) with the vector in the base frame (`attitude/baseFrameDir`). This direction of the secondary frame is the pointing direction.

   – The degree of freedom around the pointing direction is determined by the phase angle (`phaseAngle`). Two alternatives are considered for the phase angle definition:

- Providing an angle (`phaseAngle/angle`) between two directions provided by `phaseAngle/frameDir` and `phaseAngle/baseFrameDir`.

```
<frame name="secondary" baseFrame="base">
  <attitude>
    <frameDir frame="secondary"> . . . </frameDir>
    <baseFrameDir frame="base"> . . . </baseFrameDir>
    <phaseAngle>
      <frameDir frame="secondary"> . . . </frameDir>
      <baseFrameDir frame="base"> . . . </baseFrameDir>
      <angle units="deg"> . </angle>
    </phaseAngle>
  </attitude>
</frame>
```

- Providing the angle (`phaseAngle/projAngle`) between the projection in the plane perpendicular to the pointing direction of two direction provided by `phaseAngle/frameDir` and `phaseAngle/baseFrameDir`.

```
<frame name="secondary" baseFrame="base">
  <attitude>
    <frameDir frame="secondary"> . . . </frameDir>
    <baseFrameDir frame="base"> . . . </baseFrameDir>
    <phaseAngle>
      <frameDir frame="secondary"> . . . </frameDir>
      <baseFrameDir frame="base"> . . . </baseFrameDir>
      <projAngle units="deg"> . </angle>
    </phaseAngle>
  </attitude>
</frame>
```

3) 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.

4) 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 *secondary 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.

– Any attitude of a frame can initially be undefined relative to its baseframe. The attitude will become defined relative to its baseframe once its timeline element is defined.

– The attitude of any two frames is defined relative to each other if the attitude of all frames connecting the two frames in the tree is defined (see figure 5-1 below).

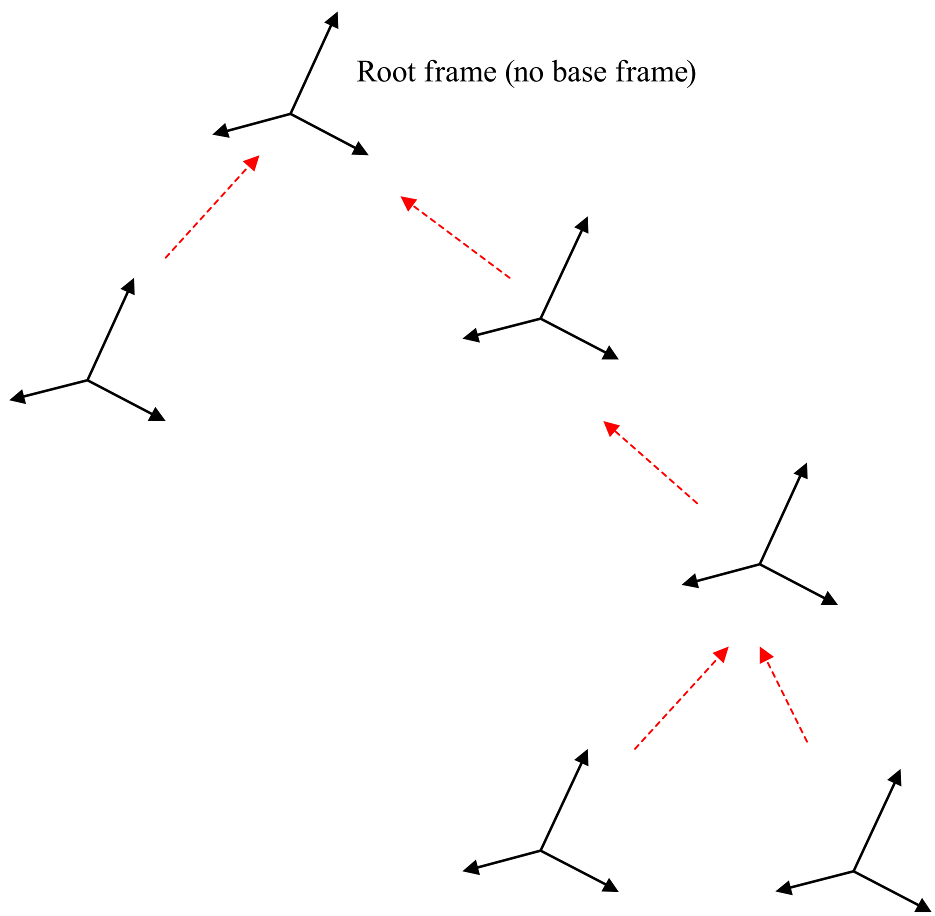

Figure 5-1: Example Tree of PRM Frames

NOTE – Dashed arrows connect every frame to its 'base frame'.

5) Direction vectors. A direction vector defined with respect to a certain frame A is implicitly defined relative to frame B, if the attitude of frame A and frame B is defined (as per bullet 4 above).

   The use of direction vectors may be constrained dependent on in which frame the vector is defined. These rules are explained in the following:

   – Vector operations can only be applied if the direction vectors are defined relative to frames that are fully define relative to one another.

   – If an attitude is defined by direction vectors (in an element of type attitude or phase angle) then the direction vector corresponding to `frameDir` and `baseFrameDir` are defined relative to frames whose attitudes are defined in their respective timelines.

The example in the figure F-2 illustrates these rules. The direction vector named vect1 is defined by a cross product of two vectors defined in frame ‘SC’ and frame ‘starTracker’. This is valid because the attitude of the frame with name starTracker is defined relative to the frame ‘SC’. If the direction vector with name ‘vect2’, was defined before the timeline element then that definition was not valid, because before the closing of the timeline element the attitude of frame ‘SC’ is not defined relative to ‘EME2000’. However, since ‘vect2’ is defined after the timeline element, by which the attitude of the frame SC becomes defined, the definition is valid.

Since the attitude of the SC is being defined in the attitude element, the direction vector corresponding to `frameDir` can be specified in the frame SC, but it could also be specified in the frame starTracker, because the frames SC and starTracker are defined relative to each other. However, it cannot be specified relative to the frame corresponding to `articulatedSolarArray`, because the attitude of that frame is not defined relative to the frame SC. The baseframe of the frame SC, is the EME2000 frame. Therefore the direction vector corresponding to the child element `baseFrame` of the `attitude` element must be defined relative to the EME2000 frame. A vector defined relative to the frames `SC`, `starTracker` or `articulatedSolarArray` would not be valid for `baseFrameDir`.

```
<definition name="%definitionName%" version="%definitionVersion%">
  <frame baseFrame="none" name="EME2000" />
  <frame baseFrame="EME2000" name="SC" />
  <frame baseFrame="SC" name="starTracker" >
    <attitude   …="">
    </frame>
  <frame name="articulatedSolarArray" baseframe”SC”=”” />
  <dirVect name=”vect1” operator=”cross”>
    <dirVect frame =”SC”> 1. 0. 0. </dirVect>
    <dirVect frame =”starTracker”> 0. 1. 0. </dirVect>
  </dirVect>
  <timeline frame=”SC”>
    <block name="inertial">
      <startEpoch localName="blockStart" />
      <endEpoch localName="blockEnd" />
      <attitude>
        <frameDir      frame="starTracker" > 1. 0. 0. </frameDir>
        <baseFrameDir frame=”EME2000” >
          0. 0. 1. </baseframeDir>
          <phaseangle …=””>
        </attitude>
    </block>
  </timeline>
  <dirVect name=”vect2” operator=”cross”>
    <dirVect frame =”SC”> 1. 0. 0. </dirVect>
    <dirVect frame =”starTracker”> 0. 1. 0. </dirVect>
  </dirVect>
</definition>
```

Figure F-2: Example of Use of Direction Vectors

The example in the figure F-3 illustrates these rules:

– The ‘SC’ frame is defined with respect to the ‘EME2000’ frame. This is the satellite attitude.

– The ‘starTracker’ frame through the orientation of a vector in the ‘starTracker’ frame with respect to the ‘SC’ frame. In the example, the X axis of both frames are aligned.

– The direction vector named vect1 is defined by a cross product of two vectors defined in frame ‘SC’ and frame ‘starTracker’. This is valid because the attitude of the frame ‘starTracker’ is defined relative to the frame ‘SC’.

– The direction vector named vect2 is defined by a cross product of two vectors defined in frame ‘SC’ and frame ‘articulatedSolarArray’. This is not valid because the attitude of the frame ‘articulatedSolarArray’ is not defined relative to the frame ‘SC’.

– The attitude within the timeline attitude block is defined through the relationship between vectors expressed in the ‘starTracker’ frame and the ‘EME2000’ frame. This is valid because the relationship between the start tracker direction and the EME2000 direction are fully defined through the relationship between the ‘EME20002 frame and the ‘SC’ frame (satellite attitude) and the relationship between the ‘starTracker’ frame and the ‘SC’ frame as defined in the ‘starTracker’ frame definition.

```
<definition name="%definitionName%" version="%definitionVersion%">
  <frame baseFrame="none" name="EME2000" />
  <frame baseFrame="EME2000" name="SC" />
  <frame baseFrame="SC" name="starTracker" >
    <attitude>
      <frameDir frame="starTracker" > 0. 0. 1. </frameDir>
      <baseFrameDir frame="SC" > 0. 0. 1. </baseFrameDir>
      <phaseAngle> ... </phaseAngle>
    </attitude >
  </frame>
  <frame name="articulatedSolarArray" baseframe="SC" />
  <dirVect name="vect1" operator="cross">
    <dirVect frame ="SC"> 1. 0. 0. </dirVect>
    <dirVect frame ="starTracker"> 0. 1. 0. </dirVect>
  </dirVect>
  <dirVect name="vect2" operator="cross">
    <dirVect frame ="SC"> 1. 0. 0. </dirVect>
    <dirVect frame =" articulatedSolarArray "> 0. 1. 0. </dirVect>
  </dirVect>
  <block name="inertial">
    <startEpoch localName="blockStart" />
    <endEpoch localName="blockEnd" />
    <attitude>
      <frameDir frame="starTracker" > 1. 0. 0. </frameDir>
      <baseFrameDir frame="EME2000" > 0. 0. 1. </baseFrameDir>
      <phaseAngle> ... </phaseAngle>
    </attitude>
  </block>
</definition>
```

Figure F-3: Example of Use of Direction Vectors

6) State Vectors, Surfaces and Surfacevectors follow the same rules as directions vectors as described in bullet 5 above.

ANNEX G

ITEMS FOR AN INTERFACE CONTROL DOCUMENT

(INFORMATIVE)

G1 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 Recommended Standard should be developed and negotiated whenever possible.

Item	**Section**
PRM transmission mechanism	1.2 3.2.1.21
Generation of PRM not covered by the standard templates	3.1
Time scales in <TIME_SYSTEM> not covered by those in annex B	3.2.1.17
Reference frames in <frameEntity> not covered by annex B	Annex B
PRM file naming convention	3.2.1.20
Spacecraft Clock time scale defined in annex B	Annex B1
Mission Elapsed Time (MET) and Mission Reference Time (MRT)	Annex B1
Data Security Implementation Specifics	E1.11

ANNEX H

ACRONYMS AND ABBREVIATIONS

(INFORMATIVE)

Acronym	Meaning
ADM	Attitude Data Messages
CCSDS	Consultative Committee for Space Data Systems
DOY	day of year
ESA	European Space Agency
ICD	interface control document
ICRF	International Celestial Reference Frame
ICS	implementation conformance statement
ID	identifier
IEEE	Institute of Electrical and Electronics Engineers
ITRF	International Terrestrial Reference Frame
JPL	Jet Propulsion Laboratory
KVN	keyword = value notation
N/A	not applicable / not available
NAIF	Navigation Ancillary Information Facility
NASA	National Aeronautics and Space Administration
NDM	Navigation Data Messages
ODM	Orbit Data Messages
OEM	Orbit Ephemeris Message
PRM	Pointing Request Message
RL	requirements list (part of ICS)
SANA	Space Assigned Numbers Authority
SC	spacecraft
SFTP	Secure File Transfer Protocol
TDM	Tracking Data Message
URL	Uniform Resource Locator
UTC	Universal Time Coordinated
XML	Extensible Markup Language
Xpath	XML Path Language
XSL	XML Stylesheet Language

ANNEX I

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 Definition 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

```xml
<metadata>
  <OBJECT_NAME>SC-Name</OBJECT_NAME>
  <OBJECT_ID>SC-ID</OBJECT_ID>
  <TIME_SYSTEM>UTC</TIME_SYSTEM>
  <START_TIME>2018-07-02T00:00:00.000</START_TIME>
  <END_TIME>2018-07-04T00:00:00.000</END_TIME>
  <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

```xml
<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 assumes 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 I-1.

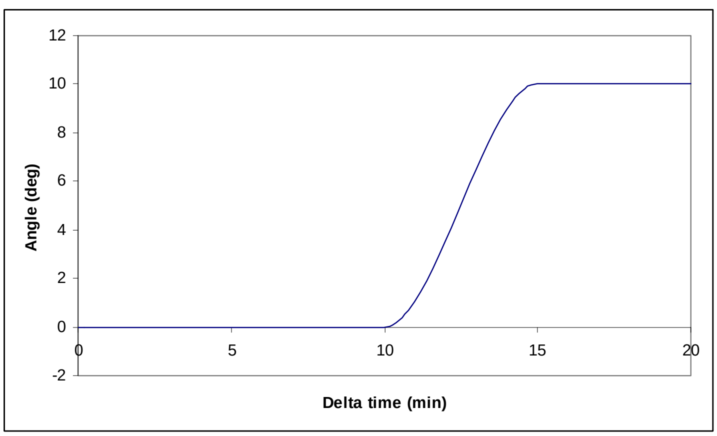

Figure I-1: Rotation Angle Versus Time

Definition	See code below

```
<metadata>
  <OBJECT_NAME>SC-Name</OBJECT_NAME>
  <OBJECT_ID>SC-ID</OBJECT_ID>
  <TIME_SYSTEM>UTC</TIME_SYSTEM>
  <START_TIME>2018-07-02T00:00:00.000</START_TIME>
  <END_TIME>2018-07-04T00:00:00.000</END_TIME>
  <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"
               units=”deg”
               frame="EME2000">270.615649 38.783689</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>
```

<table>
<tr><td>Request</td><td>
<pre>
<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>
</pre>
</td></tr>
</table>

ANNEX J

PRM TEMPLATE SAMPLES FROM PROTOTYPING

(INFORMATIVE)

This annex provides the result of the PRM prototyping activities as a reference example for the use of the PRM templates. The summary test case definition is provided for reference, followed by the instantiation of the corresponding PRM template.

<u>INERTIAL POINTING</u>

– General details:

  Root frame is GCRF
  Time scale is UTC
  Satellite is Hipparcos (1989-062B)

– The reference direction for the inertial pointing is the star Vega (RA 18h 36m 56s | Dec +38° 47′ 1″)

– The pointing axis is the spacecraft X-axis (generally towards the direction of Vega).

– The phase angle is defined such that the Y-axis is contained in the plane Vega-Spacecraft-Aldebaran (RA 4h 35m 55s | Dec +16° 30′ 33″)

– The pointing is set with an offset of 10º away from Vega to the half plane containing Aldebaran.

– The attitude is to be maintained during 3 hours after 2016-02-10T23:00:00.000

– A second pointing is to be implemented starting on 2016-02-27T00:00:00.000 for 1.5 hours. This second pointing has no offset with respect to Vega.


```xml
<?xml version="1.0" encoding="utf-8"?>
<prm xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" id="CCSDS_PRM_VERS" version="1.0">
  <header>
    <CREATION_DATE>2017-01-01T00:00:00</CREATION_DATE>
    <ORIGINATOR>NAV-WG</ORIGINATOR>
  </header>
  <body>
    &lt;segment&gt;
      <metadata>
        <TIME_SYSTEM>UTC</TIME_SYSTEM>
        <OBJECT_ID>1989-062B</OBJECT_ID>
        <OBJECT_NAME>Hipparcos</OBJECT_NAME>
        <START_TIME>2016-02-10T23:00:00</START_TIME>
        <STOP_TIME>2016-02-27T01:30:00</STOP_TIME>
        <definition name="myDefinition" version="1.0">
            <frame baseFrame="none" name="GCRF" />
            <frame baseFrame="GCRF" name="BODY" />
            <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="BODY"
                              coord="cartesian"
                              units="">0.0 1.0 0.0</frameDir>
                    <!-- Inertial reference direction for phase angle -->
                    <baseFrameDir frame="GCRF"
                                  coord="raDec"
                                  units="deg">68.97917 16.50917</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="BODY"
                              coord="cartesian"
                              units="">1.0 0.0 0.0</frameDir>
                    <!-- Inertial reference direction for offset angle -->
                    <baseFrameDir frame="GCRF"
                                  coord="raDec"
                                  units="deg">68.97917 16.50917</baseFrameDir>
                    <projAngle localName="offsetAngle" />
                </offsetAngle>
                <!-- Angular rate around the boresight -->
                <!-- Element optional; remove if no angular rate specified -->
                <angularRate localName="angularRate" />
            </attitude>
        </block>
    </definition>
  </metadata>
    <data>
        <timeline frame="BODY">
            <block ref="inertial">
                <!-- Pointing request start time -->
                <blockStart>2016-02-10T23:00:00</blockStart>
                <!-- Pointing request end time -->
                <blockEnd>2016-02-11T02:00:00</blockEnd>
                <!-- SC axis to be pointed to Target -->
                <boresight frame="BODY"
                           coord="cartesian"
                           units="">1.0 0.0 0.0</boresight>
                <!-- Inertial Target -->
                <target frame="GCRF"
                        coord="raDec"
                        units="deg">279.23334 38.78361</target>
                <!-- Phase angle, see convention in PRM annex F -->
                <!-- For spin stabilized spacecraft omit this block -->
                <phaseAngle units="deg">0.0</phaseAngle>
                <!-- Offset angle, see convention in PRM annex F -->
                <!-- Block optional; remove if no offset with respect to target -->
                <offsetAngle units="deg">10.0</offsetAngle>
                <!-- Angular rate around the boresight -->
            </block>
            <block ref="inertial">
                <!-- Pointing request start time -->
                <blockStart>2016-02-27T00:00:00</blockStart>
                <!-- Pointing request end time -->
                <blockEnd>2016-02-27T01:30:00</blockEnd>
                <!-- SC axis to be pointed to Target -->
                <boresight frame="BODY"
                           coord="cartesian"
                           units="">1.0 0.0 0.0</boresight>
                <!-- Inertial Target -->
                <target frame="GCRF"
                        coord="raDec"
                        units="deg">279.23334 38.78361</target>
                <!-- Phase angle, see convention in PRM annex F -->
                <!-- For spin stabilized spacecraft omit this block -->
```

```
                <phaseAngle units="deg">0.0</phaseAngle>
            </block>
        </timeline>
    </data>
  &lt;/segment&gt;
 </body>
</prm>
```

SUN POINTING

– General details:

  Root frame is GCRF
  Time scale is TDB
  Satellite is SOHO (1995-065A)

– The pointing axis is the spacecraft X-axis (generally towards the direction of the Sun).

– The phase angle is defined such that the Y-axis is contained in the plane Sun-Spacecraft-Aldebaran (RA 4h 35m 55s | Dec +16° 30′ 33″)

– The pointing is set with an offset of 10º away from the Sun to the half plane containing Aldebaran.

– The spacecraft shall rotate at a rate of 1º per hour around the X-axis.

– The attitude is to be maintained during 3 hours after 2016-02-10T23:00:00.000

– A second pointing is to be implemented starting on 2016-02-27T00:00:00.000 for 1.5 hours. The X-axis points to the Sun, with no offset and no rotation rate around the pointing axis.

```
<?xml version="1.0" encoding="utf-8"?>
<prm xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" id="CCSDS_PRM_VERS" version="1.0">
  <header>
    <CREATION_DATE>2017-01-01T00:00:00</CREATION_DATE>
    <ORIGINATOR>NAV-WG</ORIGINATOR>
  </header>
  <body>
    &lt;segment&gt;
        <metadata>
            <TIME_SYSTEM>TDB</TIME_SYSTEM>
            <OBJECT_ID>1995-065A</OBJECT_ID>
            <OBJECT_NAME>SOHO</OBJECT_NAME>
            <START_TIME>2016-02-10T23:00:00.000</START_TIME>
            <STOP_TIME>2016-02-27T01:30:00.000</STOP_TIME>
            <definition name="myDefinition" version="1.0">
                <frame baseFrame="none" name="GCRF" />
                <frame baseFrame="GCRF" name="BODY" />
                <frame baseFrame="none" name="BODY" />
                <block name="sunPointing">
                    <startEpoch localName="blockStart" />
                    <endEpoch localName="blockEnd" />
                    <attitude>
                        <frameDir localName="boresight" />
                        <baseFrameDir>
                            <origin>
                                <orbitFile>http://ms.esa.int/orbit/OEM_ops.xml</orbitFile>
                            </origin>
                            <target>
```

```
                    <orbit name="Sun">
                        <ephObject>SUN</ephObject>
                    </orbit>
                </target>
            </baseFrameDir>
            <!-- Phase angle provides the rotation around the boresight -->
            <!-- For spin stabilized spacecraft omit this block -->
            <phaseAngle>
                <!-- SC reference direction for phase angle -->
                <frameDir frame="BODY"
                          coord="cartesian"
                          units="">0.0 1.0 0.0</frameDir>
                <!-- Inertial reference direction for phase angle -->
                <baseFrameDir frame="GCRF"
                              coord="raDec"
                              units="deg">68.97917 16.50917</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="BODY"
                          coord="cartesian"
                          units="">0.0 1.0 0.0</frameDir>
                <!-- Inertial reference direction for offset angle -->
                <baseFrameDir frame="GCRF"
                              coord="raDec"
                              units="deg">68.97917 16.50917</baseFrameDir>
                <projAngle localName="offsetAngle" />
            </offsetAngle>
            <!-- Angular rate around the boresight -->
            <!-- Element optional; remove if no angular rate specified -->
            <angularRate localName="angularRate" />
        </attitude>
    </block>
  </definition>
</metadata>
<data>
    <timeline frame="BODY">
        <block ref="sunPointing">
            <!-- Pointing request start time -->
            <blockStart>2016-02-10T23:00:00.000</blockStart>
            <!-- Pointing request end time -->
            <blockEnd>2016-02-11T02:00:00.000</blockEnd>
            <!-- SC axis to be pointed to Target -->
            <boresight frame="BODY"
                       coord="cartesian"
                       units="">1.0 0.0 0.0</boresight>
            <!-- Phase angle provides the rotation around the boresight -->
            <!-- For spin stabilized spacecraft omit this block -->
            <phaseAngle units="deg">0.0</phaseAngle>
            <!-- Offset angle, see convention in PRM annex F -->
            <!-- Block optional; remove if no offset with respect to target -->
            <offsetAngle units="deg">10.0</offsetAngle>
            <!-- Angular rate around the boresight -->
            <!-- Element optional; remove if no angular rate specified -->
            <angularRate units="deg/s">0.277778e-3</angularRate>
        </block>
        <block ref="sunPointing">
            <!-- Pointing request start time -->
            <blockStart>2016-02-27T00:00:00.000</blockStart>
            <!-- Pointing request end time -->
            <blockEnd>2016-02-27T01:30:00.000</blockEnd>
            <!-- SC axis to be pointed to Target -->
            <boresight frame="BODY"
                       coord="cartesian"
                       units="">1.0 0.0 0.0</boresight>
            <!-- Phase angle provides the rotation around the boresight -->
            <!-- For spin stabilized spacecraft omit this block -->
            <phaseAngle units="deg">0.0</phaseAngle>
        </block>
```

```
          </timeline>
        </data>
      &lt;/segment&gt;
    </body>
  </prm>
```

TRACK WITH INERTIAL DIRECTION YAW STEERING

– General details:

  Root frame is GCRF
  Time scale is TDB
  Satellite is MEX (2003-022A)

– The spacecraft X-axis points to Mars (target direction).

– The phase angle is the angle in the plane perpendicular to the target direction of the spacecraft Y-axis from the projection of the direction to Aldebaran (RA 4h 35m 55s | Dec +16° 30′ 33″) reckoned positive around the target direction.

– The yaw steering phase angle is 0º.

```
<?xml version="1.0" encoding="utf-8"?>
<prm xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" id="CCSDS_PRM_VERS" version="1.0">
  <header>
    <CREATION_DATE>2017-02-27T17:00:00</CREATION_DATE>
    <ORIGINATOR>NAV-WG</ORIGINATOR>
  </header>
  <body>
    &lt;segment&gt;
      <metadata>
        <TIME_SYSTEM>TDB</TIME_SYSTEM>
        <OBJECT_ID>2003-022A</OBJECT_ID>
        <OBJECT_NAME>MEX</OBJECT_NAME>
        <START_TIME>2017-10-02T00:00:00.000</START_TIME>
        <STOP_TIME>2017-10-02T03:00:00.000</STOP_TIME>
        <definition name="myDefinition" version="1.0">
          <frame baseFrame="none" name="GCRF" />
          <frame baseFrame="GCRF" name="MEXFrame" />
          <orbit name="MEX">
            <!-- OEM containing the SC orbit  -->
            <orbitFile>MEX.oem.xml</orbitFile>
          </orbit>
          <orbit name="Mars">
            <!-- The following two elements cannot appear together; one must be selected -->
            <!-- Either the object name for the reference target body ...   -->
            <ephObject>Mars</ephObject>
            <!-- ... or the OEM containing the target object orbit -->
            <orbitFile>-</orbitFile>
          </orbit>
          <dirVector name="targetBody">
            <origin ref="MEX" />
            <target ref="Mars" />
          </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="MEXFrame"
```

```
                          coord="cartesian"
                          units="-">0.0 0.1 0.0</frameDir>
                <!-- Inertial reference direction for phase angle -->
                <baseFrameDir frame="GCRF"
                              coord="raDec"
                              units="deg">60.59861 16.50917</baseFrameDir>
                <projAngle localName="phaseAngle" />
              </phaseAngle>
            </attitude>
          </block>
        </definition>
      </metadata>
      <data>
        <timeline frame="MEXFrame">
          <block ref="bodyTrackWithInertialYawSteering">
            <!-- Pointing request start time -->
            <blockStart>2017-10-02T00:00:00.000</blockStart>
            <!-- Pointing request end time -->
            <blockEnd>2017-10-02T03:00:00.000</blockEnd>
            <!-- SC axis to be pointed to the target body -->
            <boresight frame="MEXFrame"
                       coord="cartesian"
                       units="-">1.0 0.0 0.0</boresight>
            <targetBody ref="Mars" />
            <!-- Phase angle, see convention in annex F-->
            <phaseAngle units="deg">0.0</phaseAngle>
          </block>
        </timeline>
      </data>
    &lt;/segment&gt;
  </body>
</prm>
```

TRACK WITH POWER OPTIMIZED YAW STEERING

– General details:

  Root frame is GCRF
  Time scale is TDB
  Satellite is MEX (2003-022A)

– The spacecraft X-axis points to Mars.

– The phase angle is defined making the spacecraft Y-axis perpendicular to the Sun direction such that the spacecraft Y-axis, the pointing direction and the Sun direction are right handed.

```xml
<?xml version="1.0" encoding="utf-8"?>
<prm xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" id="CCSDS_PRM_VERS" version="1.0">
  <header>
    <CREATION_DATE>2017-02-27T17:00:00</CREATION_DATE>
    <ORIGINATOR>NAV-WG</ORIGINATOR>
  </header>
  <body>
    &lt;segment&gt;
      <metadata>
        <TIME_SYSTEM>TDB</TIME_SYSTEM>
        <OBJECT_ID>2003-022A</OBJECT_ID>
        <OBJECT_NAME>MEX</OBJECT_NAME>
        <START_TIME>2017-10-02T00:00:00.000</START_TIME>
        <STOP_TIME>2017-10-02T03:00:00.000</STOP_TIME>
        <definition name="myDefinition" version="1.0">
          <frame baseFrame="none" name="GCRF" />
          <frame baseFrame="GCRF" name="MEXFrame" />
          <orbit name="MEX">
            <!-- OEM containing the SC orbit  -->
            <orbitFile>MEX.oem.xml</orbitFile>
          </orbit>
          <orbit name="Mars">
            <!-- The following two elements cannot appear together; one must be selected -->
            <!-- Either the object name for the reference target body ... -->
            <ephObject>Mars</ephObject>
            <!-- ... or the OEM containing the target object orbit -->
          </orbit>
          <orbit name="Sun">
            <ephObject>SUN</ephObject>
          </orbit>
          <dirVector name="targetBody">
            <origin ref="MEX" />
            <target ref="Mars" />
          </dirVector>
          <dirVector name="Sun">
            <origin ref="MEX" />
            <target ref="Sun" />
          </dirVector>
          <phaseAngle name="perpendicularToSun">
            <!-- Coordinates of SC axis to be kept perpendicular to Sun -->
            <!-- See signs convention on annex F-->
            <frameDir frame="MEXFrame"
                      coord="cartesian"
                      units="-">0.0 0.1 0.0</frameDir>
            <baseFrameDir ref="Sun" />
            <angle units="deg">90.0</angle>
          </phaseAngle>
          <block name="bodyTrackWithPowerOptimizedYawSteering">
            <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>
      <data>
        <timeline frame="MEXFrame">
          <block ref="bodyTrackWithPowerOptimizedYawSteering">
            <!-- Pointing request start time -->
            <blockStart>2017-10-02T00:00:00.000</blockStart>
            <!-- Pointing request end time -->
            <blockEnd>2017-10-02T03:00:00.000</blockEnd>
            <!-- SC axis to be pointed to the target body -->
            <boresight frame="MEXFrame"
                       coord="cartesian"
                       units="-">1.0 0.0 0.0</boresight>
          </block>
        </timeline>
      </data>
    &lt;/segment&gt;
  </body>
</prm>
```

NADIR WITH POWER OPTIMIZED YAW STEERING

– General details:

  Root frame is ITRF
  Time scale is UTC
  Satellite is METOP-1A (2006-044A)

– The central body used to define the nadir direction is the Earth.

– The spacecraft Z-axis points to the nadir direction.

– The phase angle is defined making the spacecraft Y-axis perpendicular to the Sun direction such that the spacecraft Y-axis, the pointing direction and the Sun direction are right handed.


```xml
<?xml version="1.0" encoding="utf-8"?>
<prm xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" id="CCSDS_PRM_VERS" version="1.0">
  <header>
    <CREATION_DATE>2017-02-27T17:00:00</CREATION_DATE>
    <ORIGINATOR>NAV-WG</ORIGINATOR>
  </header>
  <body>
    &lt;segment&gt;
      <metadata>
        <TIME_SYSTEM>UTC</TIME_SYSTEM>
        <OBJECT_ID>2006-044A</OBJECT_ID>
        <OBJECT_NAME>METOP-1A</OBJECT_NAME>
        <START_TIME>2017-10-02T00:00:00.000</START_TIME>
        <STOP_TIME>2017-10-02T03:00:00.000</STOP_TIME>
        <definition name="myDefinition" version="1.0">
          <frame baseFrame="none" name="ITRF" />
          <frame baseFrame="ITRF" name="MetopFrame" />
          <orbit name="METOP-1A">
            <!-- OEM containing the SC orbit  -->
            <orbitFile>metop_1a.oem.xml</orbitFile>
          </orbit>
          <orbit name="Sun">
            <ephObject>SUN</ephObject>
          </orbit>
          <frame baseFrame="ITRF" name="EARTH">
            <attitude>
              <!-- Planet reference frame -->
              <rotation from="ITRF" to="ITRF" />
            </attitude>
          </frame>
          <orbit name="EARTH">
            <!-- Object name for the planet  -->
            <ephObject>EARTH</ephObject>
          </orbit>
          <surface name="nadirReferenceSurface" frame="EARTH">
            <origin ref="EARTH" />
            <!-- Planet reference ellipsoid -->
            <a units="km">6378.137</a>
            <b units="km">6356.752</b>
          </surface>
          <dirVector name="Sun">
            <origin ref="METOP-1A" />
            <target ref="Sun" />
          </dirVector>
          <dirVector name="nadir">
            <origin ref="METOP-1A" />
            <target>
              <surfaceVector operator="normal">
                <surface ref="nadirReferenceSurface" />
```

```
              <origin ref="METOP-1A" />
            </surfaceVector>
          </target>
        </dirVector>
        <phaseAngle name="perpendicularToSun">
          <!-- Coordinates of SC axis to be kept perpendicular to Sun -->
          <!-- See signs convention on annex F-->
          <frameDir frame="MetopFrame"
                    coord="cartesian"
                    units="-">0.0 1.0 0.0</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>
    <data>
      <timeline frame="MetopFrame">
        <block ref="nadir">
          <!-- Pointing request start time -->
          <blockStart>2017-10-02T00:00:00.000</blockStart>
          <!-- Pointing request end time -->
          <blockEnd>2017-10-02T03:00:00.000</blockEnd>
          <!-- SC axis to be pointed to Nadir -->
          <boresight frame="MetopFrame"
                     coord="cartesian"
                     units="-">0.0 0.0 1.0</boresight>
        </block>
      </timeline>
    </data>
  &lt;/segment&gt;
</body>
</prm>
```

NADIR WITH GROUNDTRACK ALIGNED YAW STEERING

– General details:

  Root frame is ITRF
  Time scale is UTC
  Satellite is METOP-1A (2006-044A)

– The central body used to define the nadir direction is the Earth.

– The spacecraft Z-axis points to the nadir direction.

– The Y-axis points perpendicular to the plane defined by the nadir direction and the tangent to the ground track.[3] The spacecraft Y-axis, the nadir direction and the tangent in direction of increasing time shall form a right handed coordinate system.

```
<?xml version="1.0" encoding="utf-8"?>
<prm xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" id="CCSDS_PRM_VERS" version="1.0">
  <header>
    <CREATION_DATE>2017-02-27T17:00:00</CREATION_DATE>
    <ORIGINATOR>NAV-WG</ORIGINATOR>
  </header>
  <body>
    &lt;segment&gt;
      <metadata>
        <TIME_SYSTEM>UTC</TIME_SYSTEM>
        <OBJECT_ID>2006-044A</OBJECT_ID>
        <OBJECT_NAME>METOP-1A</OBJECT_NAME>
        <START_TIME>2017-10-02T00:00:00.000</START_TIME>
        <STOP_TIME>2017-10-02T03:00:00.000</STOP_TIME>
        <definition name="myDefinition" version="1.0">
          <frame baseFrame="none" name="ITRF" />
          <frame baseFrame="ITRF" name="MetopFrame" />
          <orbit name="METOP-1A">
            <!-- OEM containing the SC orbit  -->
            <orbitFile>metop_1a.oem.xml</orbitFile>
          </orbit>
          <frame baseFrame="ITRF" name="EARTH">
            <attitude>
              <!-- Planet reference frame -->
              <rotation from="ITRF" to="ITRF" />
            </attitude>
          </frame>
          <orbit name="EARTH">
            <!-- Object name for the planet  -->
            <ephObject>EARTH</ephObject>
          </orbit>
          <surface name="nadirReferenceSurface" frame="EARTH">
            <origin ref="EARTH" />
            <!-- Planet reference ellipsoid  -->
            <a units="km">6378.137</a>
            <b units="km">6356.752</b>
          </surface>
          <dirVector name="nadir">
            <origin ref="METOP-1A" />
            <target>
              <surfaceVector name="groundTrack" operator="normal">
                <surface ref="nadirReferenceSurface" />
                <origin ref="METOP-1A" />
              </surfaceVector>
            </target>
```

[3] 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.

```
          </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 F-->
            <frameDir frame="MetopFrame"
                      coord="cartesian"
                      units="-">0.0 1.0 0.0</frameDir>
            <baseFrameDir ref="tangent" />
            <angle units="deg">90.0</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="perpendicularToGroundTrack" />
            </attitude>
          </block>
        </definition>
      </metadata>
      <data>
        <timeline frame="MetopFrame">
          <block ref="nadir">
            <!-- Pointing request start time -->
            <blockStart>2017-10-02T00:00:00.000</blockStart>
            <!-- Pointing request end time -->
            <blockEnd>2017-10-02T03:00:00.000</blockEnd>
            <!-- SC axis to be pointed to Nadir -->
            <boresight frame="MetopFrame"
                       coord="cartesian"
                       units="-">0.0 0.0 0.1</boresight>
          </block>
        </timeline>
      </data>
    &lt;/segment&gt;
  </body>
</prm>
```

NADIR WITH ORBITAL POLE ALIGNED YAW STEERING

– General details:

  Root frame is GCRF
  Time scale is TDB
  Satellite is ULYSSES (1990-090B)

– The central body used to define the nadir direction is Jupiter.

– The spacecraft Z-axis points to the nadir direction.

– The spacecraft Y-axis points in the direction of the projection of the orbit pole (direction of the orbit angular momentum) in the plane perpendicular to the nadir.

```
<?xml version="1.0" encoding="utf-8"?>
<prm xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" id="CCSDS_PRM_VERS" version="1.0">
  <header>
    <CREATION_DATE>2017-02-27T17:00:00</CREATION_DATE>
    <ORIGINATOR>NAV-WG</ORIGINATOR>
  </header>
  <body>
```

```
&lt;segment&gt;
  <metadata>
    <TIME_SYSTEM>TDB</TIME_SYSTEM>
    <OBJECT_ID>1990-090B</OBJECT_ID>
    <OBJECT_NAME>ULYSSES</OBJECT_NAME>
    <START_TIME>2017-10-02T00:00:00.000</START_TIME>
    <STOP_TIME>2017-10-02T30:00:00.000</STOP_TIME>
    <definition name="myDefinition" version="1.0">
      <frame baseFrame="none" name="GCRF" />
      <frame baseFrame="GCRF" name="ULYFrame" />
      <orbit name="ULYSSES">
        <!-- OEM containing the SC orbit  -->
        <orbitFile>ulysses.oem.xml</orbitFile>
      </orbit>
      <frame baseFrame="GCRF" name="Jupiter">
        <attitude>
          <!-- Planet reference frame -->
          <rotation from="GCRF" to="JUP2000" />
        </attitude>
      </frame>
      <orbit name="Jupiter">
        <!-- Object name for the planet  -->
        <ephObject>Jupiter</ephObject>
      </orbit>
      <surface name="nadirReferenceSurface" frame="Jupiter">
        <origin ref="Jupiter" />
        <!-- Planet reference ellipsoid  -->
        <a units="km">69911.0</a>
        <b units="km">69911.0</b>
      </surface>
      <dirVector name="nadir">
        <origin ref="ULYSSES" />
        <target>
          <surfaceVector operator="normal">
            <surface ref="nadirReferenceSurface" />
            <origin ref="ULYSSES" />
          </surfaceVector>
        </target>
      </dirVector>
      <dirVector name="orbitalPole" operator="cross">
        <dirVector name="scToTargetBody">
          <origin ref="ULYSSES" />
          <target ref="Jupiter" />
        </dirVector>
        <dirVector ref="scToTargetBody" operator="derivative" />
      </dirVector>
      <phaseAngle name="alignedWithOrbitalPole">
        <!-- Coordinates of SC axis to be kept aligned with the orbit pole -->
        <!-- See signs convention in annex F -->
        <frameDir frame="ULYFrame"
                  coord="cartesian"
                  units="-">0.0 1.0 0.0</frameDir>
        <baseFrameDir ref="orbitalPole" />
        <angle units="deg">0.0</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>
  <data>
    <timeline frame="ULYFrame">
      <block ref="nadir">
        <!-- Pointing request start time -->
        <blockStart>2017-10-02T00:00:00.000</blockStart>
```

```
          <!-- Pointing request end time -->
          <blockEnd>2017-10-02T30:00:00.000</blockEnd>
          <!-- SC axis to be pointed to Nadir -->
          <boresight frame="ULYFrame"
                     coord="cartesian"
                     units="-">0.0 0.0 1.0</boresight>
        </block>
      </timeline>
    </data>
  &lt;/segment&gt;
 </body>
</prm>
```

LIMB POINTING WITH POWER OPTIMIZED YAW STEERING

– General details:

  Root frame is ITRF
  Time scale is UTC
  Satellite is METOP-1A (2006-044A)

– The central body used to define the limb direction is the Earth.

– The spacecraft Z-axis points to a limb direction 10 km above the Earth surface.

– The limb point direction uses as inertial reference direction Aldebaran (RA 4h 35m 55s | Dec +16° 30′ 33″)

– The phase angle is defined making the spacecraft Y-axis perpendicular to the Sun direction such that the spacecraft Y-axis, the pointing direction and the Sun direction are right handed.

```
<?xml version="1.0" encoding="utf-8"?>
<prm xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" id="CCSDS_PRM_VERS" version="1.0">
  <header>
    <CREATION_DATE>2017-01-01T00:00:00.000</CREATION_DATE>
    <ORIGINATOR>NAV-WG</ORIGINATOR>
  </header>
  <body>
    &lt;segment&gt;
      <metadata>
        <TIME_SYSTEM>UTC</TIME_SYSTEM>
        <OBJECT_ID>2006-044A</OBJECT_ID>
        <OBJECT_NAME>METOP-1A</OBJECT_NAME>
        <START_TIME>2017-10-02T23:00:00</START_TIME>
        <STOP_TIME>2017-10-02T23:00:00</STOP_TIME>
        <definition name="myDefinition" version="1.0">
          <frame baseFrame="none" name="ITRF2000" />
          <frame baseFrame="ITRF2000" name="SC_BODY_FRAME" />
          <orbit name="METOP-1A">
            <!-- OEM containing the SC orbit  -->
            <orbitFile>oem.xml</orbitFile>
          </orbit>
          <orbit name="EARTH">
            <!-- Object name for the reference target body  -->
            <ephObject>EARTH</ephObject>
          </orbit>
          <orbit name="Sun">
            <ephObject>SUN</ephObject>
          </orbit>
          <frame baseFrame="ITRF2000" name="EARTH">
            <attitude>
              <!-- Planet reference frame -->
              <rotation from="ITRF2000" to="ITRF2000" />
```

```
            </attitude>
          </frame>
          <surface name="limbReferenceSurface" frame="EARTH">
            <origin ref="EARTH" />
            <!-- Planet reference ellipsoid  -->
            <a units="km">6376.136</a>
            <b units="km">6256.345</b>
          </surface>
          <dirVector name="Sun">
            <origin ref="METOP-1A" />
            <target ref="Sun" />
          </dirVector>
          <phaseAngle name="perpendicularToSun">
            <!-- Coordinates of SC axis to be kept perpendicular to Sun -->
            <!-- See signs convention on annex F-->
            <frameDir frame="SC_BODY_FRAME"
                      coord="cartesian"
                      units="">0.0 1.0 0.0</frameDir>
            <baseFrameDir ref="Sun" />
            <angle units="deg"> 90. </angle>
          </phaseAngle>
          <block name="limbWithPowerOptimizedYawSteering">
            <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>
      <data>
        <timeline frame="SC_BODY_FRAME">
          <block ref="limbWithPowerOptimizedYawSteering">
            <!-- Pointing request start time -->
            <blockStart>2017-10-02T23:00:00</blockStart>
            <!-- Pointing request end time -->
            <blockEnd>2017-10-03T23:00:00</blockEnd>
            <!-- SC axis to be pointed to the target body -->
            <boresight frame="SC_BODY_FRAME"
                       coord="cartesian"
                       units="">0.0 0.0 1.0</boresight>
            <target>
              <surface ref="limbReferenceSurface" />
              <dirVector coord="raDec" units="deg">68.979167 16.509167</dirVector>
              <height units="km">10.0</height>
            </target>
          </block>
        </timeline>
      </data>
    &lt;/segment&gt;
  </body>
</prm>
```

LIMB POINTING WITH INERTIAL DIRECTION YAW STEERING

– General details:

  Root frame is ITRF
  Time scale is UTC
  Satellite is METOP-1A (2006-044A)

– The central body used to define the limb direction is the Earth.

– The spacecraft Z-axis points to a limb direction 10 km above the Earth surface.

– The limb point direction uses as inertial reference direction Aldebaran (RA 4h 35m 55s | Dec +16° 30′ 33″)

– The phase angle is the angle in the plane perpendicular to the target direction of the spacecraft Y-axis from the projection of the direction to Aldebaran (RA 4h 35m 55s | Dec +16° 30′ 33″) reckoned positive around the target direction.

– The yaw steering phase angle is 45º.


```
<?xml version="1.0" encoding="utf-8"?>
<prm xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" id="CCSDS_PRM_VERS" version="1.0">
  <header>
    <CREATION_DATE>2017-02-27T17:00:00</CREATION_DATE>
    <ORIGINATOR>NAV-WG</ORIGINATOR>
  </header>
  <body>
    &lt;segment&gt;
      <metadata>
        <TIME_SYSTEM>UTC</TIME_SYSTEM>
        <OBJECT_ID>2006-044A</OBJECT_ID>
        <OBJECT_NAME>METOP-1A</OBJECT_NAME>
        <START_TIME>2017-02-27T00:00:00.000</START_TIME>
        <STOP_TIME>2017-02-27T30:00:00.000</STOP_TIME>
        <definition name="myDefinition" version="1.0">
          <frame baseFrame="none" name="ITRF" />
          <frame baseFrame="ITRF" name="SCFrame" />
          <orbit name="METOP-1A">
            <!-- OEM containing the SC orbit  -->
            <orbitFile>metop_1a_oem.xml</orbitFile>
          </orbit>
          <orbit name="Earth">
            <!-- Object name for the reference target body  -->
            <ephObject>Earth</ephObject>
          </orbit>
          <frame baseFrame="ITRF" name="Earth">
            <attitude>
              <!-- Planet reference frame -->
              <rotation from="ITRF" to="ITRF" />
            </attitude>
          </frame>
          <surface name="limbReferenceSurface" frame="Earth">
            <origin ref="Earth" />
            <!-- Planet reference ellipsoid  -->
            <a units="km">6378.137</a>
            <b units="km">6340.000</b>
          </surface>
          <!-- Inertial reference direction for phase angle -->
          <phaseAngle>
            <!-- SC reference direction for phase angle -->
            <frameDir frame="SCFrame"
                      coord="cartesian"
```

```
                    units="-">0.0 1.0 0.0</frameDir>
          <!-- Inertial reference direction for phase angle -->
          <baseFrameDir frame="ITRF"
                        coord="raDec"
                        units="deg">68.9792 16.50916667</baseFrameDir>
          <projAngle localName="phaseAngle" />
        </phaseAngle>
        <block name="limbWithInertialPointingYawSteering">
          <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>
    <data>
      <timeline frame="SCFrame">
        <block ref="limbWithInertialPointingYawSteering">
          <!-- Pointing request start time -->
          <blockStart>2017-02-27T00:00:00.000</blockStart>
          <!-- Pointing request end time -->
          <blockEnd>2017-02-27T03:00:00.000</blockEnd>
          <!-- SC axis to be pointed to the target body -->
          <boresight frame="SCFrame"
                     coord="cartesian"
                     units="-">0.0 0.0 0.1</boresight>
          <target>
            <surface ref="limbReferenceSurface" />
            <dirVector coord="raDec" units="deg">60.59861111 16.50916667</dirVector>
            <height units="km">10</height>
          </target>
          <!-- Phase angle, see convention in annex F-->
          <phaseAngle units="deg">45.0</phaseAngle>
        </block>
      </timeline>
    </data>
  &lt;/segment&gt;
 </body>
</prm>
```

VELOCITY POINTING WITH ORBITAL POLE YAW STEERING

– General details:

  Root frame is GCRF
  Time scale is TDB
  Satellite is BEPI-COLOMBO (2018-013A)

– The central body used to define the pointing is Mercury.

– The spacecraft X-axis is pointed towards the spacecraft velocity relative to Mercury.

– The spacecraft Y-axis points in the direction of the projection of the orbit pole (direction of the orbit angular momentum) in the plane perpendicular to the nadir.

```xml
<?xml version="1.0" encoding="utf-8"?>
<prm xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" id="CCSDS_PRM_VERS" version="1.0">
  <header>
    <CREATION_DATE>2017-02-27T17:00:00</CREATION_DATE>
    <ORIGINATOR>NAV-WG</ORIGINATOR>
  </header>
  <body>
    &lt;segment&gt;
      <metadata>
        <TIME_SYSTEM>TDB</TIME_SYSTEM>
        <OBJECT_ID>2018-013A</OBJECT_ID>
        <OBJECT_NAME>BEPI-COLOMBO</OBJECT_NAME>
        <START_TIME>2017-02-10T00:00:00.000</START_TIME>
        <STOP_TIME>2017-02-10T30:00:00.000</STOP_TIME>
        <definition name="myDefinition" version="1.0">
          <frame baseFrame="none" name="GCRF" />
          <frame baseFrame="GCRF" name="BCFrame" />
          <orbit name="BEPI-COLOMBO">
            <!-- OEM containing the SC orbit  -->
            <orbitFile>bp_oem.xml</orbitFile>
          </orbit>
          <orbit name="Mercury">
            <!-- Object name for the reference target body  -->
            <ephObject>Mercury</ephObject>
          </orbit>
          <dirVector name="velocity" operator="derivative">
            <origin ref="Mercury" />
            <target ref="BEPI-COLOMBO" />
          </dirVector>
          <dirVector name="position">
            <origin ref="Mercury" />
            <target ref="BEPI-COLOMBO" />
          </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="BCFrame"
                          coord="cartesian"
                          units="-">0.0 1.0 0.0</frameDir>
                <!-- Reference direction for phase angle -->
```

```
              <baseFrameDir ref="orbitalPole" />
              <projAngle localName="phaseAngle" />
            </phaseAngle>
          </attitude>
        </block>
      </definition>
    </metadata>
    <data>
      <timeline frame="BCFrame">
        <block ref="velocityWithOrbitalPoleYawSteering">
          <!-- Pointing request start time -->
          <blockStart>2017-02-10T00:00:00.000</blockStart>
          <!-- Pointing request end time -->
          <blockEnd>2017-02-10T30:00:00.000</blockEnd>
          <!-- SC axis to be pointed in the direction of the relative velocity -->
          <boresight frame="BCFrame"
                     coord="cartesian"
                     units="-">1.0 0.0 0.0</boresight>
          <phaseAngle units=“deg">0.0</phaseAngle>
        </block>
      </timeline>
    </data>
  &lt;/segment&gt;
</body>
</prm>
```