



European Space Research
and Technology Centre
Keplerlaan 1
2201 AZ Noordwijk
The Netherlands
T +31 (0)71 565 6565
F +31 (0)71 565 6040
www.esa.int

DOCUMENT

OPS Angle Computation using the EO Mission Software CFI

Prepared by Maurizio De Bartolomei
Reference PE-TN-ESA-GS-344
Issue 1
Revision 0
Date of Issue 28/09/2012
Status Issued
Document Type TN
Distribution



APPROVAL

Title OPS Angle Computation using the EO Mission Software CFI	
Issue 1	Revision 0
Author M. De Bartolomei	Date 28/09/2012
Approved by	Date
M. Zundo B. Duesmann	28/09/2012

CHANGE LOG

Reason for change	Issue	Revision	Date
First Issue	1	0	28/9/2012

CHANGE RECORD

Issue 1	Revision 0		
Reason for change	Date	Pages	Paragraph(s)
First Issue	28/9/2012	All	All



Table of contents:

1	INTRODUCTION	4
1.1	Applicable Documents	4
1.2	Reference Documents	4
2	COMPUTATION OVERVIEW.....	4
3	COMPUTATION STEPS	5

1 INTRODUCTION

Purpose of this document is to describe how to compute the OPS angle using the Earth Observation Mission Software Branch 4.x (see http://eop-cfi.esa.int/CFI/eo_cfi_software.html).

The OPS angle is defined in [OPS_DEF].

The implementation of this computation in the Mission Software is consistent with the description given in section 4.3.1 of [OPS_ALGO] and is available as per version 4.4.

1.1 Applicable Documents

[OPS_DEF] “Earth Observation OPS commanding”,
PE-TN-ESA-SY-0305, Version 1.9, 11-11-2011

1.2 Reference Documents

[OPS_ALGO] “OPS Angle Definition & Calculation,
PE-TN-ESA-SY-0338, Issue 1.2, 25-09-2012

[ORB_UM] “Orbit Library User Manual“, Issue 4.4
http://eop-cfi.esa.int/CFI/EO_CFI_DOCS/4.4/C_Libraries/SUMs/OrbitSUM_v4_4.pdf

[LIB_UM] “Lib Library User Manual“, Issue 4.4
http://eop-cfi.esa.int/CFI/EO_CFI_DOCS/4.4/C_Libraries/SUMs/LibSUM_v4_4.pdf

2 COMPUTATION OVERVIEW

Fig. 1 shows the computation flow.

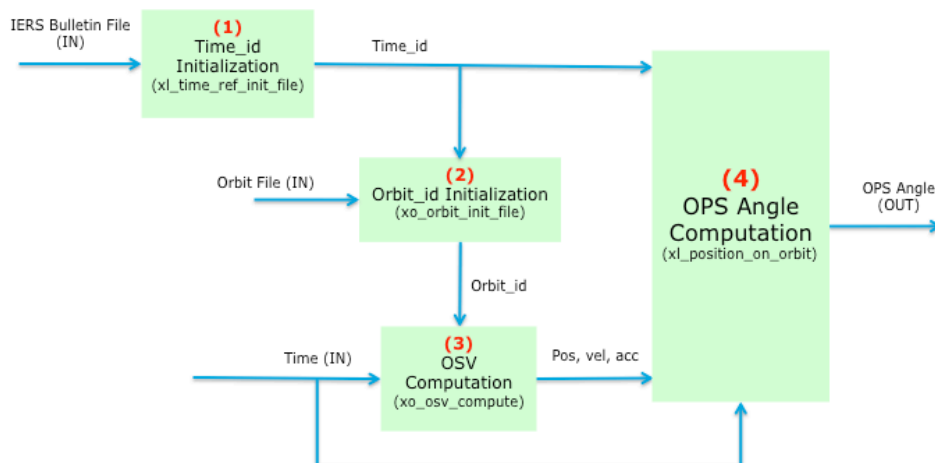


Fig. 1: computation flow.



For the computation of the OPS angle the following inputs are required:

1. **A `time_id` properly initialized.** The `time_id` is an object storing data (i.e. time correlations and polar motion information) necessary for accurate coordinate system conversions. The polar motion information can be stored only if the `time_id` is initialized using IERS bulletin A and/or B file. This is the recommended method for initialization in order to allow the most accurate conversion from Earth Fixed to an inertial frame. In the following it is assumed that the `time_id` is initialized using such method.
2. **An Orbit State Vector (OSV).** This is the satellite position, velocity and acceleration expressed in the Earth Fixed reference system at a given time. This information (position, velocity acceleration and time) may be retrieved from an external source or can be computed with the Mission Software itself. In the following it is assumed that the State Vector is computed with the Mission Software. For the computation of the the Orbit State Vector the following inputs are required:
 - a. **An `orbit_id` properly initialized.** The `orbit_id` is an object storing information required to compute a State Vector. There are several methods to initialize an `orbit_id` and they are described in [ORB_UM]. However in the following it is assumed that the `orbit_id` is initialized using an orbit file.
 - b. **The input time.** The input time needs to be expressed in processing format, however several functions are available to convert from other formats (e.g. calendar/ASCII or transport), for more details see [LIB_UM].

3 COMPUTATION STEPS

The following description makes reference to the Mission Software C API. However, equivalent methods exist in the Java and C++ APIs.

- 1) **`time_id` initialization:** call function `x1_time_ref_init_file` (see section 7.1 of [LIB_UM]). Inputs shall be set according to section 7.1.3 of [LIB_UM], in particular:
 - a. the variables `n_files` and `time_file` shall be set to the input bulletin file(s).
- 2) **`orbit_id` initialization:** call function `xo_orbit_init_file` (see section 7.5 of [ORB_UM]). Inputs shall be set according to section 7.5.3 of [ORB_UM], in particular:
 - a. the variables `n_files` and `time_file` shall be set to the input orbit file(s)
 - b. the input `time_id` shall be the one initialized in step 1.
- 3) **Orbit State Vector (OSV) computation:** call function `xo_osv_compute` (see section 7.33 of [ORB_UM]). Inputs shall be set according to section 7.33.3 of [ORB_UM], in particular:
 - a. variables `time` and `time_ref` shall be set to the input time
 - b. the input `orbit_id` shall be the one initialized in step 2.



- 4) **OPS angle computation:** call function **xl_position_on_orbit** (see section 7.41 of [LIB_UM]). Inputs shall be set according to section 7.41.3 of [LIB_UM], in particular:
- the **time_id** shall be the one initialized in step 1
 - the orbit state vector (time, pos, vel and acc variables) shall set to be the ones computed in step 3
 - angle_type** shall be set to **XL_ANGLE_TYPE_TRUE_LAT_EF**.

The output variable **angle** in step 4 is the requested OPS angle.

Notes:

- The sequence above refers to the computation of a single OPS angle. In case more OPS angles need to be computed (i.e. the OPS angle needs to be computed for a list of times), steps 1 and 2 need to be executed only once.
- The functions above also require a **model_id** object. This can be initialized using the function **xl_model_init** and **mode** set to **XL_MODEL_DEFAULT** (see [LIB_UM]).