RE-ENGINEERING OF MISSION ANALYSIS SOFTWARE FOR ENVISAT-1

PPF_LIB SOFTWARE USER MANUAL

PO-IS-DMS-GS-0557

Code:	PO-IS-DMS-GS-0557
Issue:	5.9
Date:	30/05/11

	Name	**Function**	**Signature**
Prepared by:	Mariano Sánchez Nogales	Project Engineer	
Checked by:	José Antonio González Abeytua	Project Manager	
Approved by:	José Antonio González Abeytua	Project Manager	

DEIMOS Space S.L.U.
Ronda de Poniente, 19
Edificio Fiteni VI, Portal 2, 2ª Planta
28760 Tres Cantos(Madrid), SPAIN
Tel.: +34 91 806 34 50
Fax: +34 91 806 34 51
E-mail: deimos@deimos-space.com

Document Information

Contract Data		Classification	
Contract Number:	Contract number	Internal	
		Public	
Contract Issuer:	ESA / ESTEC	Industry	X
		Confidential	

Internal Distribution		
Name	Unit	Copies

External Distribution		
Name	Organisation	Copies

Archiving	
Word Processor:	Framemaker 6.0
File name:	rgt-dms-sum-002-10
Archive Code	P/SUM/DMS/01/043-015

Document Status Log

Issue	Change Description	Date	Approval
4.1	• First Version	15/01/97	
4.2	• `int` variables replaced by `long` • variables expressed in Mean of Date CS now in True of Date CS • pl_sbtutc and pl_utcsbt functions use pointers in all their variables • known problems of release 4.1 solved	30/04/97	
4.3	• runtime performances improved • pl_geo_distance algorithm accuracy improved • known problems of release 4.2 solved	18/05/98	
4.4	• error handling vectors length modified • port to Windows	19/10/98	
4.5	• thresholds used to increase speed updated (see section 6.1) • error handling modified in the following functions: - pl_car_geo - pl_emjd - pl_geo_distance - pl_sbtutc - pl_utcsbt • DUT1 range updated to [-1.0,1.0]	31/05/99	
4.6	• error handling modified in the following functions: - pl_tsub - pl_geo_distance • DUT1 range change (-0.999999, 0.999999) • pl_geo_distance SPR 54 (antipodal points) solved	14/04/00	
4.7	• Unreleased	22/06/01	
4.8	• Unreleased	31/07/01	
4.9	• Unreleased • New document from previous version by: - L. J. Alvarez (GMV) - F. J. Jimenez (GMV) - J. A. Gonzalez (GMV) - E. Lopez (GMV) - M. Sanchez-Nogales (GMV)	22/10/01	
5.0	• Document format change.	18/01/02	
5.1	• Unreleased	25/11/02	
5.2	• See change bars	26/05/03	
5.3	• Maintenance Release	13/12/04	

Issue	Change Description	Date	Approval
5.3.1	• Maintenance Release	15/02/05	
5.4	• Maintenance Release	17/05/05	
5.5	• Maintenance Release • Library Optimization	30/01/06	
5.6	• Maintenance Release • New version for LINUX 64-Bits	22/12/06	
5.7	• Changes for Envisat extended life • New version for MAC OS on Intel platforms • Maintenance Release	11/04/08	
5.8	• Maintenance release.	15/09/09	
5.9	• Maintenance release.	30/05/11	

Table of Contents

1 SCOPE

The Software User Manual (SUM) of the Envisat-1 mission CFI software is composed of

- a general document describing the sections common to all the CFI software libraries
- a specific document for each of those libraries.

This document is the PPF_LIB Software User Manual. It provides a detailed description of the use of the CFI functions included within the PPF_LIB CFI software library.

2 ACRONYMS AND NOMENCLATURE

2.1 Acronyms

AOCS	Attitude and Orbit Control System
ANX	Ascending Node Crossing
CFI	Customer Furnished Item
CS	Coordinate System
DRS	Data Relay Satellite
ESA	European Space Agency
ESTEC	European Space Technology and Research Centre
FOS	Flight Operation Segment
GS	Ground Station
H/W	Hardware
I/F	Interface
LOS	Line Of Sight
OBT	On-board Binary Time
PPF	Polar Platform
RAM	Random Access Memory
SBT	Satellite Binary Time
SRAR	Satellite Relative Actual Reference
SSP	Sub Satellite Point
SUM	Software User Manual
S/W	Software
UTC	Universal Time Coordinated
UT1	Universal Time UT1
WGS[84]	World Geodetic System 1984

2.2 Nomenclature

CFI	A group of CFI functions, and related software and documentation, that will be distributed by ESA to the users as an independent unit
CFI function	A single function within a CFI that can be called by the user
Library	A software library containing all the CFI functions included within a CFI plus the supporting functions used by those CFI functions (transparently to the user)

3 APPLICABLE AND REFERENCE DOCUMENTS

3.1 Applicable documents

AD 1 Finalization of the re-engineering of Mission Analysis Software and of the ROP Generation Tool for Envisat: Statement of Work.PO-SW-ESA-SY-1242. ESA/ESTEC/APP. Issue 1.1. 03/10/2001.

AD 2 ESA Software Engineering Standards. ESA PSS-05-0. ESA. Issue 2. February 1991

3.2 Reference documents

RD 1 Envisat-1 Mission CFI Software Description and Interface Definition Document. PO-ID-ESA-SY-00412

RD 2 Envisat-1 Mission CFI Software. Mission Conventions Document. PO-IS-GMV-GS-0561

RD 3 Envisat-1 Mission CFI Software General Software User Manual. PO-IS-DMS-GS-0556

RD 4 Envisat-1 Products Specifications. PO-RS-MDA-GS-2009

4 INTRODUCTION

This software library contains all low-level generic routines, supporting all the other CFI functions.

The following CFI functions are included:

- **pl_change_sv_cs**: transforms a state vector between different coordinate systems.
- **pl_car_geo**: transforms from cartesian to geodetic coordinates.
- **pl_geo_car**: transforms from geodetic to cartesian coordinates.
- **pl_srar_cs**: calculates the Satellite Relative Actual Reference coordinate system and the rotation angles between this coordinate system and the Satellite Reference one.
- **pl_geo_distance**: calculates the geodesic distance between two points that lay on the same ellipsoid, and the azimuth of the related geodesic line at both points.
- **pl_tmjd**: changes the format in which a UT1 time is expressed from Transport format to Processing and External format.
- **pl_pmjd**: changes the format in which a UT1 time is expressed from Processing format to External and Transport format.
- **pl_emjd**: changes the format in which a UT1 time is expressed form External format to Processing and Transport format.
- **pl_tadd**: adds two UT1 times expressed in Transport format
- **pl_tsub**: substracts two UT1 times expressed in Transport format
- **pl_sbtutc**: transforms from SBT to UTC.
- **pl_utcsbt**: transforms from UTC to SBT.
- **pl_sun**: calculates the position and velocity of the Sun in the True of Date coordinate system
- **pl_moon**: calculates the Moon position and velocity in the True of Date coordinate system
- **pl_planets**: calculates the position and velocity of a selected planet in the Heliocentric Mean of 2000.0 coordinate system
- **pl_star_radec**: calculates the right ascension and declination of a star in the True of Date coordinate system.

A detailed description of each function is provided in section 7.

Please refer also to:

- RD 2 for a detailed description of the time references and formats, coordinate systems, parameters and models used in this document
- RD 3 for a complete overview of the CFI, and in particular the detailed description of the error handling functions

5 LIBRARY INSTALLATION

For a detailed description of the installation of any CFI library, please refer to RD 3.

6 LIBRARY USAGE

To use the PPF_LIB software library in a user application, that application must include in his source code either:

- `ppf_lib.h` (for a C application)
- `ppf_lib.inc` (for a Fortran application under SOLARIS/AIX/LINUX/MacOS)
- `ppf_lib_win.inc` (for a Fortran application under Windows 95/NT)

To link correctly this application, the user must include in his linking command flags like (assuming *cfi_libs_dir* and *cfi_include_dir* are the directories where respectively all CFI libraries and include files have been installed, see RD 3 for installation procedures):

- SOLARIS / AIX

  `-I`*cfi_include_dir* `-L`*cfi_lib_dir* `-lppf_lib`
- WINDOWS

  `/I` *"cfi_include_dir"* /libpath:*"cfi_lib_dir"* `libppf_lib.lib`

All functions described in this document have a name starting with the prefix `pl_`

To avoid problems in linking a user application with the `PPF_LIB` software library due to the existence of names multiple defined, the user application should avoid naming any global software item beginning with either the prefix `PL_` or `pl_`.

To preserve compatibility with the historical CFI function names, it is possible to call the CFI functions described in this document from a user application with or without the `pl_` prefix. This does not apply to new CFI functions (e.g. coordinate systems functions) or to the error handling functions, which are described in the General SUM (see RD 3).

This is summarized in the table below.

Function Name	Enumeration value	Long
Main CFI Functions		
pl_change_sv_cs	PL_CHANGE_SV_CS_ID	0
pl_car_geo	PL_CAR_GEO_ID	1
pl_geo_car	PL_GEO_CAR_ID	2
pl_srar_cs	PL_SRAR_CS_ID	3
pl_geo_distance	PL_GEO_DISTANCE_ID	4
pl_tmjd tmjd	PL_TMJD_ID	5
pl_emjd emjd	PL_EMJD_ID	6
pl_pmjd pmjd	PL_PMJD_ID	7
pl_tadd tadd	PL_TADD_ID	8
pl_tsub tsub	PL_TSUB_ID	9
pl_sbtutc sbtutc	PL_SBTUTC_ID	10

Function Name	Enumeration value	Long
pl_utcsbt utcsbt	PL_UTCSBT_ID	11
pl_sun	PL_SUN_ID	12
pl_moon	PL_MOON_ID	13
pl_planets	PL_PLANETS_ID	14
pl_star_radec	PL_STAR_RADEC_ID	15
Error Handling Functions		
pl_verbose	not applicable	
pl_silent		
pl_vector_code		
pl_vector_msg		
pl_print_msg		

Notes about the table:

- to transform the extended status flag returned by a CFI function to either a list of error codes or list of error messages, the enumeration value (or the corresponding long value) described in the table must be used
- the error handling functions have no enumerated values

Whenever available **it is strongly recommended to use enumeration values rather than integer values**.

6.1 Usage hints

The runtime performances of few of the CFI functions are improved to a large extent if they are called two consecutive times keeping constant some of their inputs:

- `pl_change_sv_cs`:UT1 time
- `pl_srar_cs`: position, velocity and acceleration vectors, aocs, mispointing and mispointing rate.
- `pl_planets`: UTC time
- `pl_star_radec`: UTC time

In fact, the UT1 time, position, velocity and acceleration vectors do not need to keep exactly constant as long as the difference between two consecutive calls lays within the following thresholds:

- UT1/UTC time: 0.0864 microsec
- Position vector: 0.6e-3 m
- Velocity vector: 0.6e-6 m/s
- Acceleration vector: 0.6e-9 m/s^2
- AOCS: 5e-9 deg
- Mispointing angles: 5e-9 deg
- Mispointing angles-rate: 5e-12 deg

Furthermore, the same runtime improvement is achieved in other CFI functions that, although the user may not need to call two consecutive times with the same inputs, are called internally by other higher level CFI functions in those conditions, and thus improving the runtime performances of the latter.

This is the case of the following CFI functions:

- `pl_car_geo`: position, velocity and acceleration vectors
- `pl_sun`: UTC time
- `pl_moon`: UTC time

7 CFI FUNCTIONS DESCRIPTION

The following sections describe each CFI function.

The calling interfaces are described both for C users and Fortran users.

Input and output parameters of each CFI function are described in tables, where C programming language syntax is used to specify:

- parameter types (e.g. long, double)
- array sizes of N elements (e.g. param[N])
- array element M (e.g. [M])

Fortran users should adapt the tables using Fortran syntax equivalent terms:

- parameter types (e.g. long <=> INTEGER*4, double <=>REAL*8)
- array sizes of N elements (e.g. param[N] <=> param (N))
- array element M (e.g. [M] <=> (M+1))

7.1 pl_change_sv_cs

7.1.1 Overview

The **pl_change_sv_cs** CFI function transforms a state vector between different coordinate systems.

7.1.2 Calling interface

The calling interface of the **pl_change_sv_cs** CFI function is the following (input parameters are underlined):

```
#include <ppf_lib.h>
{
       double time[2];
       long pcs_in, pcs_out;
       double r[3], rd[3], r2d[3];
       double r_out[3], rd_out[3], r2d_out[3];
       long status;

       status = pl_change_sv_cs (time, &pcs_in, &pcs_out,
                                 r, rd, r2d,
                                 r_out, rd_out, r2d_out);
}
```

For Fortran programs the declaration and calling procedure is as follows (input parameters are underlined, note that the C preprocessor must be used because of the presence of the `#include` statement):

```
#include <ppf_lib.inc>
       REAL*8 TIME(2)
       INTEGER*4 PCS_IN, PCS_OUT
       INTEGER*4 STATUS
       REAL*8 R(3),RD(3),R2D(3)
       REAL*8 R_OUT(3),RD_OUT(3),R2D_OUT(3)

       STATUS = PL_CHANGE_SV_CS (TIME, PCS_IN, PCS_OUT,
      &                          R, RD, R2D,
      &                          R_OUT, RD_OUT, R2D_OUT)
```

7.1.3 Input parameters

The **pl_change_sv_cs** CFI function has the following input parameters:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
time[2]	double	[0]	Reference UTC (UT1 time)	decimal days (Processing format)	>= -18262 < +36525
		[1]	ΔUT1 = UT1 - UTC (UT1 time)	s (Processing format)	>= -1.0 <= +1.0
pcs_in	long *	-	Initial coordinate system ID	-	1 - 6
pcs_out	long *	-	Final coordinate system ID	-	1 - 6
r[3]	double	all	Input position vector (Initial coordinate system)	m	-
rd[3]	double	all	Input velocity vector (Initial coordinate system)	m/s	-
r2d[3]	double	all	Input acceleration vector (Initial coordinate system)	m/s^2	-

It is possible to use enumeration values rather than integer values for some of the input arguments:

Input	Description	Enumeration value	Long
pcs_in	Barycentric Mean of 2000.0 CS	PL_BM2000	1
	Heliocentric Mean of 2000.0 CS	PL_HM2000	2
	Mean of 2000.0 CS	PL_GM2000	3
	Mean of Date CS	PL_MOD	4
	True of Date CS	PL_TOD	5
	Earth fixed CS	PL_EF	6
pcs_out	Barycentric Mean of 2000.0 CS	PL_BM2000	1
	Heliocentric Mean of 2000.0 CS	PL_HM2000	2
	Mean of 2000.0 CS	PL_GM2000	3
	Mean of Date CS	PL_MOD	4
	True of Date CS	PL_TOD	5
	Earth fixed CS	PL_EF	6

7.1.4 Output parameters

The output parameters of the **pl_change_sv_cs** CFI function are:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
pl_change_sv_cs	long	-	Extended status flag	-	-
r_out[3]	double	all	Output position vector (Final coordinate system)	m	-
rd_out[3]	double	all	Output velocity vector (Final coordinate system)	m/s	-
r2d_out[3]	double	all	Output acceleration vector (Final coordinate system)	m/s^2	-

7.1.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **pl_change_sv_cs** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the PPF_LIB software library **pl_vector_msg** (see RD 3).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the extended status flag returned by the **pl_change_sv_cs** function by calling the function of the PPF_LIB software library **pl_vector_code** (see RD 3)

Error type	Error message	Cause and impact	Error code	Error No
ERR	Wrong UTC on input (out of range)	No calculation performed	PL_CFI_E1_NDAYS_ERR	0
ERR	Wrong DUT1 on input (out of range)	No calculation performed	PL_CFI_E1_DUT1_ERR	1
ERR	Wrong initial coordinate system on input (not allowed)	No calculation performed	PL_CFI_INPUT_CS_ERR	2
ERR	Wrong final coordinate system on input (not allowed)	No calculation performed	PL_CFI_OUTPUT_CS_ERR	3

7.1.6 Runtime performances

The following runtime performances have been measured.

Two runtimes are provided, one with fixed inputs, i.e. the function has been called several times with the same UT1 time, but modifying the other input parameters; and a second one with random inputs, i.e all the inputs have been modified from call to call and the average time has been taken.

Ultra Sparc [ms] RANDOM inputs	Ultra Sparc [ms] FIXED inputs
0.222	0.114

7.2 pl_car_geo

7.2.1 Overview

The **pl_car_geo** CFI function transforms from cartesian to geodetic coordinates.

7.2.2 Calling interface

The calling interface of the **pl_car_geo** function is the following (input parameters are underlined):

```
#include <ppf_lib.h>
{
      double r[3], rd[3];
      double lon, lat, h, lond, latd, hd;
      long status;

      status = pl_car_geo (r, rd, &lon, &lat, &h, &lond, &latd, &hd);
}
```

For Fortran programs the declaration and calling procedure is as follows (input parameters are underlined, note that the C preprocessor must be used because of the presence of the `#include` statement):

```
#include <ppf_lib.inc>

      REAL*8 R(3),RD(3)
      REAL*8 LON, LAT, H, LOND, LATD, HD
      INTEGER*4 STATUS

      STATUS = PL_CAR_GEO (R, RD, LON, LAT, H, LOND, LATD, HD)
```

7.2.3 Input parameters

The **pl_car_geo** CFI function has the following input parameters:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
r[3]	double	all	Cartesian position vector (Earth fixed CS)	m	$r > a_{WGS} - b_{WGS}$
rd[3]	double	all	Cartesian velocity vector (Earth fixed CS)	m/s	-

7.2.4 Output parameters

The output parameters of the **pl_car_geo** CFI function are:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
pl_car_geo	long	-	Extended status flag	-	-
lon	double *	-	Geocentric longitude (Earth fixed CS)	deg	>= 0 < +360
lat	double *	-	Geodetic latitude (Earth fixed CS)	deg	>= -90 <= +90
h	double *	-	Geodetic altitude (Earth fixed CS)	m	-
lond	double *	-	Geocentric longitude rate (Earth fixed CS)	deg/s	-
latd	double *	-	Geodetic latitude rate (Earth fixed CS)	deg/s	-
hd	double *	-	Geodetic altitude rate (Earth fixed CS)	m/s	-

7.2.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **pl_car_geo** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the PPF_LIB software library **pl_vector_msg** (see RD 3).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the extended status flag returned by the **pl_car_geo** function by calling the function of the PPF_LIB software library **pl_vector_code** (see RD 3)

Error type	Error message	Cause and impact	Error code	Error No
WARN	Geocentric longitude set to 0 deg (ambiguous case)	The geodetic latitude is +/- 90 deg, and the longitude cannot be determined (ambiguous)	PL_CFI_AMBIGUITY_CASE_WARN	0
ERR	Input vector out of valid range	Calculation not performed	PL_CFI_WRONG_INPUT_VECTOR_ERR	2

7.2.6 Runtime performances

The following runtime performances have been measured.

Ultra Sparc [ms]
0.155

7.3 pl_geo_car

7.3.1 Overview

The **pl_geo_car** CFI function transforms from geodetic to cartesian coordinates.

7.3.2 Calling interface

The calling interface of the **pl_geo_car** CFI function is the following (input parameters are underlined):

```
#include <ppf_lib.h>
{
       double r[3], rd[3];
       double lon, lat, h, lond, latd, hd;
       long status;

       status = pl_geo_car (&lon, &lat, &h, &lond, &latd, &hd, r, rd);

}
```

For Fortran programs the declaration and calling procedure is as follows (input parameters are underlined, note that the C preprocessor must be used because of the presence of the `#include` statement):

```
#include <ppf_lib.inc>

       REAL*8 R(3),RD(3)
       REAL*8 LON, LAT, H, LOND, LATD, HD
       INTEGER*4 STATUS

       STATUS = PL_GEO_CAR (LON, LAT, H, LOND, LATD, HD, R, RD)
```

7.3.3 Input parameters

The **pl_geo_car** CFI function has the following input parameters:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
lon	double *	-	Geocentric longitude (Earth fixed CS)	deg	>= 0 < 360
lat	double *	-	Geodetic latitude (Earth fixed CS)	deg	>= -90 <= +90
h	double *	-	Geodetic altitude (Earth fixed CS)	m	h >= $-b_{WGS}$
lond	double *	-	Geocentric longitude rate (Earth fixed CS)	deg/s	-
latd	double *	-	Geodetic latitude rate (Earth fixed CS)	deg/s	-
hd	double *		Geodetic altitude rate (Earth fixed CS)	m/s	-

7.3.4 Output parameters

The output parameters of the **pl_geo_car** CFI function are:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
pl_geo_car	long	-	Extended status flag	-	-
r[3]	double	all	Cartesian position vector (Earth fixed CS)	m	-
rd[3]	double	all	Cartesian velocity vector (Earth fixed CS)	m/s	-

7.3.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **pl_geo_car** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the PPF_LIB software library **pl_vector_msg** (see RD 3).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the extended status flag returned by the **pl_geo_car** function by calling the function of the PPF_LIB software library **pl_vector_code** (see RD 3)

Error type	Error message	Cause and impact	Error code	Error No
ERR	Wrong geodetic latitude on input (out of range)	No computation performed	PL_CFI_EL_GT_90_ERR	0

The altitude of the geodetic state vector is not checked, so in case it does not satisfy its allowed range it may result in raising an internal error (see 8).

7.3.6 Runtime performances

The following runtime performances have been measured.

Ultra Sparc [ms]
0.038

7.4 pl_srar_cs

7.4.1 Overview

The **pl_srar_cs** CFI function calculates the Satellite Relative Actual Reference coordinate system and the rotation angles between this coordinate system and the Satellite Reference one.

7.4.2 Calling interface

The calling interface of the **pl_srar_cs** CFI function is the following (input parameters are underlined):

```
#include <ppf_lib.h>
{
       double r[3], rd[3], r2d[3];
       double aocs[3], misp[3], mispd[3];
       double xs[3], ys[3], zs[3];
       double xsd[3], ysd[3], zsd[3];
       double angout[3], angoutd[3];
       long status;
       status = pl_srar_cs (r, rd, r2d, aocs, misp, mispd,
                            xs, ys, zs, xsd,  ysd,  zsd,
                               angout, angoutd);
}
```

For Fortran programs the declaration and calling procedure is as follows (input parameters are underlined, note that the C preprocessor must be used because of the presence of the `#include` statement):

```
#include <ppf_lib.inc>

       REAL*8 R(3),RD(3), R2D(3)
       REAL*8 AOCS(3), MISP(3), MISPD(3)
       REAL*8 XS(3), YS(3), ZS(3)
       REAL*8 XSD(3), YSD(3), ZSD(3)
       REAL*8 ANGOUT(3), ANGOUTD(3)
       INTEGER*4 STATUS
       STATUS = PL_SRAR_CS (R, RD, R2D, AOCS, MISP, MISPD,
     &                        XS, YS, ZS, XSD, YSD, ZSD,
     &                         ANGOUT, ANGOUTD)
```

7.4.3 Input parameters

The **pl_srar_cs** CFI function has the following input parameters:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
r[3]	double	all	Satellite position vector (True of Date CS)	m	-
rd[3]	double	all	Satellite velocity vector (True of Date CS)	m/s	-
r2d[3]	double	all	Satellite acceleration vector (True of Date CS)	m/s^2	-
aocs[3]	double	[0]	AOCS Cx parameter [pitch] (Satellite Reference CS)	deg	-
		[1]	AOCS Cy parameter [roll] (Satellite Reference CS)	deg	-
		[2]	AOCS Cz parameter [yaw] (Satellite Reference CS)	deg	-
misp[3]	double	[0]	Pitch mispointing angle (Satellite Relative Reference CS)	deg	-
		[1]	Roll mispointing angle (Satellite Relative Reference CS)	deg	-
		[2]	Yaw mispointing angle (Satellite Relative Reference CS)	deg	-
mispd[3]	double	[0]	Pitch mispointing rate (Satellite Relative Reference CS)	deg/s	-
		[1]	Roll mispointing rate (Satellite Relative Reference CS)	deg/s	-
		[2]	Yaw mispointing rate (Satellite Relative Reference CS)	deg/s	-

7.4.4 Output parameters

The output parameters of the **pl_srar_cs** CFI function are:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
pl_srar_cs	long	-	Extended status flag	-	-
xs[3]	double	all	Unitary direction vector along the x-axis of SRAR CS (True of Date CS)	-	-
ys[3]	double	all	Unitary direction vector along the y-axis of SRAR CS (True of Date CS)	-	-
zs[3]	double	all	Unitary direction vector along the z-axis of SRAR CS (True of Date CS)	-	-
xsd3]	double	all	First time derivative of xs (True of Date CS)	1/s	-
ysd[3]	double	all	First time derivative of ys (True of Date CS)	1/s	-
zsd[3]	double	all	First time derivative of zs (True of Date CS)	1/s	-
angout[3]	double	[0]	Satellite pitch angle (Satellite Reference CS)	deg	>= -180 < +180
		[1]	Satellite roll angle (Satellite Reference CS)	deg	>= -180 < +180
		[2]	Satellite yaw angle (Satellite Reference CS)	deg	>= -180 < +180
angoutd[3]	double	[0]	Satellite pitch rate (Satellite Reference CS)	deg/s	-
		[1]	Satellite roll rate (Satellite Reference CS)	deg/s	-
		[2]	Satellite yaw rate (Satellite Reference CS)	deg/s	-

To transform a unitary direction vector expressed in the SRAR coordinate system u_{SRAR} into a unitary direction vector expressed in the True of Date coordinate system u_{TOD}, the following expressions should be used:

$$\bar{u}_{TOD} = \begin{bmatrix} x_s[0] & y_s[0] & z_s[0] \\ x_s[1] & y_s[1] & z_s[1] \\ x_s[2] & y_s[2] & z_s[2] \end{bmatrix} \bar{u}_{SRAR}$$

$$\dot{\bar{u}}_{TOD} = \begin{bmatrix} x_s[0] & y_s[0] & z_s[0] \\ x_s[1] & y_s[1] & z_s[1] \\ x_s[2] & y_s[2] & z_s[2] \end{bmatrix} \dot{\bar{u}}_{SRAR} + \begin{bmatrix} \dot{x}_s[0] & \dot{y}_s[0] & \dot{z}_s[0] \\ \dot{x}_s[1] & \dot{y}_s[1] & \dot{z}_s[1] \\ \dot{x}_s[2] & \dot{y}_s[2] & \dot{z}_s[2] \end{bmatrix} \bar{u}_{SRAR}$$

7.4.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **pl_srar_cs** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the PPF_LIB software library **pl_vector_msg** (see RD 3).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the extended status flag returned by the **pl_srar_cs** function by calling the function of the PPF_LIB software library **pl_vector_code** (see RD 3)

Error type	Error message	Cause and impact	Error code	Error No
ERR	SRAR coordinate system cannot be defined (position and velocity vectors are zero or parallel)	Calculation not performed	PL_CFI_SRR_UNDEFINED_ERR	0
WARN	Roll and Yaw angles cannot be determined (ambiguous)	Pitch angle is +/- 90 deg, so the roll and yaw angles cannot be determined (ambiguous)	PL_CFI_PITCH_90DEG_WARN	2

7.4.6 Runtime performances

The following runtime performances have been measured.

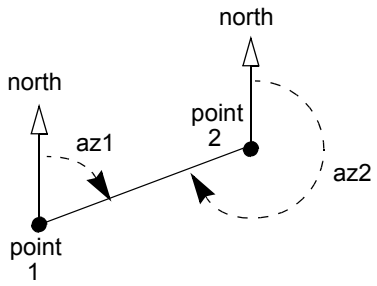
Two runtimes are provided, one with fixed inputs, i.e. the function has been called several times with the same position, velocity and acceleration vectors, but modifying the other input parameters; and a second one with random inputs, i.e all the inputs have been modified from call to call and the average time has been taken.

Ultra Sparc [ms] RANDOM inputs	Ultra Sparc [ms] FIXED inputs
0.369	0.050

7.5 pl_geo_distance

7.5.1 Overview

The **pl_geo_distance** CFI function calculates the geodesic distance between two points that lay on the same ellipsoid, and the azimuth of the related geodesic line at both points. See diagram below.



7.5.2 Calling interface

The calling interface of the **pl_geo_distance** CFI function is the following (input parameters are underlined):

```
#include <ppf_lib.h>
{
      double lon1, lat1, lon2, lat2, h;
      double d, az1, az2;
      long status;
      status = pl_geo_distance (&lon1, &lat1, &lon2, &lat2, &h,
                                &d, &az1, &az2);
}
```

For Fortran programs the declaration and calling procedure is as follows (input parameters are underlined, note that the C preprocessor must be used because of the presence of the `#include` statement):

```
#include <ppf_lib.inc>

      REAL*8 LON1, LAT1, LON2, LAT2, H
      REAL*8 D, AZ1, AZ2
      INTEGER*4 STATUS

      STATUS = PL_GEO_DISTANCE (LON1, LAT1, LON2, LAT2, H,
     &                          D, AZ1, AZ2)
```

7.5.3 Input parameters

The **pl_geo_distance** CFI function has the following input parameters:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
lon1	double *	-	Geocentric longitude of the first input point (Earth fixed CS)	deg	>= 0 < 360
lat1	double *	-	Geodetic latitude of the first input point (Earth fixed CS)	deg	>= -90 <= +90
lon2	double *	-	Geocentric longitude of the second input point (Earth fixed CS)	deg	>= 0 < 360
lat2	double *	-	Geodetic latitude of the sec-ond input point (Earth fixed CS)	deg	>= -90 <= +90
h	double *	-	Geodetic altitude of both input points (Earth fixed CS)	m	h >= $-b_{WGS}$

7.5.4 Output parameters

The output parameters of the **pl_geo_distance** CFI function are:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
pl_geo_distance	long	-	Extended status flag	-	-
d	double *	-	Geodesic distance between the two input points (Earth fixed CS)	m	>= 0
az1	double *	-	Azimuth of the geodesic line from point 1 to point 2 (Topocentric CS)	deg	>= 0 < 360
az2	double *	-	Azimuth of the geodesic line from point 2 to point 1 (Topocentric CS) Note that az2 = az1 + 180 approximately	deg	>= 0 < 360

7.5.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **pl_geo_distance** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the PPF_LIB software library **pl_vector_msg** (see RD 3).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the extended status flag returned by the **pl_geo_distance** function by calling the function of the PPF_LIB software library **pl_vector_code** (see RD 3)

Error type	Error message	Cause and impact	Error code	Error No
ERR	Different altitudes in the two points	No calculation performed	PL_CFI_ALT_NOT_ID_ERR	0
WARN	Antipodal points.Two possible azimuth values (0 or 180). Selected value is 0.0 deg.	Calculation performed Message to inform the user.	PL_CFI_ANTIPODAL_POINTS_WARN	1
ERR	Calculation not performed in PL_Geo_Car	No calculation performed	PL_CFI_GEO_CAR_ERR	2
ERR	Calculation not performed in PL_Pt_Dir_Range	No calculation performed	PL_CFI_PT_DIR_RANGE_ERR	3
ERR	No solution returned by PL_Dir_Pointing	No calculation performed	PL_CFI_ELLIPSOID_PLANE_ERR	4
ERR	Default values returned by PL_Dir_Pointing	No calculation performed	PL_CFI_DIR_POINT_WARN	5

Note that the warning No.1 is new.

The altitude of the two points is not checked, so in case it does not satisfy its allowed range it may result in raising an internal error (see 8).

If antipodal points are entered, a new warning message (number 1) is raised.

For antipodal points, a little variation of the input coordinates may lead to incoherent values for the output distance, depending on the point location on the ellipsoid.

7.5.6 Runtime performances

The following runtime performances have been measured.

Ultra Sparc [ms]
0.468

7.6 pl_tmjd

7.6.1 Overview

The **pl_tmjd** CFI function changes the format in which a UT1 time is expressed from Transport format to Processing and External format.

7.6.2 Calling interface

The calling interface of the **pl_tmjd** CFI function is the following (input parameters are underlined):

```
#include <ppf_lib.h>
{
       double mjdp[2];
       long status, mjdt[4];
       char utce[28], dut1e[9];

       status = pl_tmjd (mjdt, mjdp, utce, dut1e);
}
```

For Fortran programs the declaration and calling procedure is as follows (input parameters are underlined, note that the C preprocessor must be used because of the presence of the `#include` statement):

```
#include <ppf_lib.inc>

       REAL*8 MJDP(2)
       INTEGER*4 STATUS, MJDT(4)
       CHARACTER UTCE(28), DUT1E(9)
       STATUS = PL_TMJD(MJDT, MJDP, UTCE, DUT1E)
```

7.6.3 Input parameters

The **pl_tmjd** CFI function has the following input parameters:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
mjdt[4]	long	[0]	UTC (UT1 time)	integer days (Transport format)	>= -18262 <= +36524
		[1]		integer seconds (Transport format)	>= 0 < 86400
		[2]		integer μs (Transport format)	>= 0 <= 999999
		[3]	ΔUT1 = UT1 - UTC (UT1 time)	integer μs (Transport format)	>= -999999 <= +999999

7.6.4 Output parameters

The **pl_tmjd** CFI function has the following output parameters:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
pl_tmjd	long	-	Extended status flag	-	-
mjdp[2]	double	[0]	UTC (UT1 time)	decimal days (Processing format)	>= -18262.0 < +36525.0
		[1]	ΔUT1 = UT1 - UTC (UT1 time)	s (Processing format)	>= -1.0 <= +1.0
utce	char[28]	all	UTC (UT1 time)	(External format)	>= 01-JAN-1950 00:00:00.000000 <= 31-DEC-2099 23:59:59.999999
dut1e	char[9]	all	ΔUT1 = UT1 - UTC (UT1 time)	s (External format)	>= -0.999999 <= +0.999999

7.6.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **pl_tmjd** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the PPF_LIB software library **pl_vector_msg** (see RD 3).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the extended status flag returned by the **pl_tmjd** CFI function by calling the function of the PPF_LIB software library **pl_vector_code** (see RD 3).

Error type	Error message	Cause and impact		Error No
ERR	Input NUMBER of DAYS is out of limits (-18262,36524)	No calculation performed	PL_CFI_K19_NDAYS_ERR	0
ERR	Input SECONDS are out of limits (0-86399)	No calculation performed	PL_CFI_K19_NSECS_ERR	1
ERR	Input MICROSECONDS are out of limits (0-999999)	No calculation performed	PL_CFI_K19_NMICS_ERR	2
ERR	Input DELTA UT1 is out of limits (+-999999)	No calculation performed	PL_CFI_K19_NDUT1_ERR	3

7.6.6 Runtime performances

The following runtime performances have been measured.

Ultra Sparc [ms]
0.035

7.7 pl_pmjd

7.7.1 Overview

The **pl_pmjd** CFI function changes the format in which a UT1 time is expressed from Processing format to Transport and External format.

7.7.2 Calling interface

The calling interface of the **pl_pmjd** CFI function is the following (input parameters are underlined):

```
#include <ppf_lib.h>
{
       double mjdp[2];
       long status, mjdt[4];
       char utce[28], dut1e[9];

       status = pl_pmjd(mjdt, mjdp, utce, dut1e);
}
```

For Fortran programs the declaration and calling procedure is as follows (input parameters are underlined, note that the C preprocessor must be used because of the presence of the `#include` statement):

```
#include <ppf_lib.inc>

       REAL*8 MJDP(2)
       INTEGER*4 STATUS, MJDT(4)
       CHARACTER UTCE(28), DUT1E(9)

       STATUS = PL_PMJD(MJDT, MJDP, UTCE, DUT1E)
```

7.7.3 Input parameters

The **pl_pmjd** CFI unction has the following input parameters:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
mjdp[2]	double	[0]	UTC (UT1 time)	decimal days (Processing format)	>= -18262.0 < +36525.0
		[1]	ΔUT1 = UT1 - UTC (UT1 time)	s (Processing format)	>= -1.0 <= +1.0

7.7.4 Output parameters

The output parameters of the **pl_pmjd** CFI function are:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
pl_pmjd	long	-	Extended status flag	-	-
mjdt[4]	long	[0]	UTC (UT1 time)	integer days (Transport format)	>= -18262 <= +36524
		[1]		integer seconds (Transport format)	>= 0 < 86400
		[2]		integer μs (Transport format)	>= 0 <= 999999
		[3]	ΔUT1 = UT1 - UTC (UT1 time)	integer μs (Transport format)	>= -999999 <= +999999
utce	char[28]	all	UTC (UT1 time)	(External format)	>= 01-JAN-1950 00:00:00.000000 <= 31-DEC-2099 23:59:59.999999
dut1e	char[9]	all	ΔUT1 = UT1 - UTC (UT1 time)	s (External format)	>= -0.999999 <= +0.999999

7.7.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **pl_pmjd** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the PPF_LIB software library **pl_vector_msg** (see RD 3).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the extended status flag returned by the **pl_pmjd** CFI function by calling the function of the PPF_LIB software library **pl_vector_code** (see RD 3).

Error type	Error message	Cause and impact		Error No
ERR	Input NUMBER of DAYS is out of limits (-18262,36525)	No calculation performed	PL_CFI_K20_NDAYS_ERR	0
ERR	Input DELTA UT1 is out of limits (-1.0, 1.0)	No calculation performed	PL_CFI_K20_DUT1_ERR	1

7.7.6 Runtime performances

The following runtime performances have been measured.

Ultra Sparc [ms]
0.078

7.8 pl_emjd

7.8.1 Overview

The **pl_emjd** CFI function changes the format in which a UT1 time is expressed from External format to Transport and Processing format.

7.8.2 Calling interface

The calling interface of the **pl_emjd** CFI function is the following (input parameters are underlined):

```
#include <ppf_lib.h>
{
       double mjdp[2];
       long status, mjdt[4];
       char utce[28],dut1e[9];

       status = pl_emjd(mjdt, mjdp, utce, dut1e);
}
```

For Fortran programs the declaration and calling procedure is as follows (input parameters are underlined, note that the C preprocessor must be used because of the presence of the `#include` statement):

```
#include <ppf_lib.inc>

       REAL*8 MJDP(2)
       INTEGER*4 MJDT(4), STATUS
       CHARACTER UTCE(28),DUT1E(9)

       STATUS = PL_EMJD (MJDT, MJDP, UTCE, DUT1E)
```

7.8.3 Input parameters

The **pl_emjd** CFI function has the following input parameters:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
utce	char[28]	all	UTC (UT1 time)	(External format)	>= 01-JAN-1950 00:00:00.000000 <= 31-DEC-2099 23:59:59.999999
dut1e	char[9]	all	ΔUT1 = UT1 - UTC (UT1 time)	s (External format)	>= -0.999999 <= +0.999999

7.8.4 Output parameters

The output parameters of the **pl_emjd** CFI function are:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
pl_emjd	long	-	Extended status flag	-	-
mjdt[4]	long	[0]	UTC (UT1 time)	integer days (Transport format)	>= -18262 <= +36524
		[1]		integer seconds (Transport format)	>= 0 < 86400
		[2]		integer μs (Transport format)	>= 0 <= 999999
		[3]	ΔUT1 = UT1 - UTC (UT1 time)	integer μs (Transport format)	>= -999999 <= +999999
mjdp[2]	double	[0]	UTC (UT1 time)	decimal days (Processing format)	>= -18262.0 < +36525.0
		[1]	ΔUT1 = UT1 - UTC (UT1 time)	s (Processing format)	>= -1.0 <= +1.0

7.8.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **pl_emjd** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the PPF_LIB software library **pl_vector_msg** (see RD 3).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the extended status flag returned by the **pl_emjd** function by calling the function of the PPF_LIB software library **pl_vector_code** (see RD 3)

Error type	Error message	Cause and impact	Error code	Error No
ERR	Input EXTERNAL FORMAT DATE STRING is not correct	No calculation performed	PL_CFI_K21_MICS_ERR	0
ERR	Input MONTH STRING is not correct	No calculation performed	PL_CFI_K21_MONTH_ERR	1
ERR	Invalid Gregorian calendar date (hour, minutes or seconds out of range)	No calculation performed	PL_CFI_K21_CAL_ERR	2
ERR	Invalid Gregorian calendar date	No calculation performed	PL_CFI_K21_CAL2_ERR	3
ERR	Input DELTA UT1 is not correct	No calculation performed	PL_CFI_K21_DUT1_ERR	4

Note that some error numbers have changed since previous version, while keeping the error name.

7.8.6 Runtime performances

The following runtime performances have been measured.

Ultra Sparc [ms]
0.151

7.9 pl_tadd

7.9.1 Overview

The **pl_tadd** CFI function adds two UT1 times expressed in Transport format

7.9.2 Calling interface

The calling interface of the **pl_tadd** CFI function is the following (input parameters are underlined):

```
#include <ppf_lib.h>
{
      long mjdt[4], mjdrt[4], addt[4], status;

      status = pl_tadd(mjdt, mjdrt, addt);
}
```

For Fortran programs the declaration and calling procedure is as follows (input parameters are underlined, note that the C preprocessor must be used because of the presence of the `#include` statement):

```
#include <ppf_lib.inc>

      INTEGER*4 MJDT(4), MJDRT(4), ADDT(4), STATUS

      STATUS = PL_TADD (MJDT, MJDRT, ADDT)
```

7.9.3 Input parameters

The **pl_tadd** CFI function has the following input parameters:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
mjdt[4]	long	[0]	UTC (UT1 time)	integer days (Transport format)	>= -18262 <= +36524
		[1]		integer seconds (Transport format)	>= 0 < 86400
		[2]		integer μs (Transport format)	>= 0 <= 999999
		[3]	ΔUT1 = UT1 - UTC (UT1 time)	integer μs (Transport format)	>= -999999 <= +999999
addt[4]	long	[0]	UTC to be added (UT1 time)	integer days (Transport format)	
		[1]		integer seconds (Transport format)	
		[2]		integer μs (Transport format)	
		[3]	ΔUT1 to be added (UT1 time) Important note: unless exceptional circumstances ΔUT1 values should not be added, thus this parameter should be zero	integer μs (Transport format)	

7.9.4 Output parameters

The output parameters of the **pl_tadd** CFI function are:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
pl_tadd	long	-	Extended status flag	-	-
mjdrt[4]	long	[0]	Resulting UTC (UT1 time)	integer days (Transport format)	>= -18262 <= +36524
		[1]		integer seconds (Transport format)	>= 0 < 86400
		[2]		integer μs (Transport format)	>= 0 <= 999999
		[3]	Resulting ΔUT1 (UT1 time)	integer μs (Transport format)	>= -1999999 <= +1999999

Note that, for some cases, the resulting ΔUT1 may exceed the maximum value of 0.999999 seconds allowed for the external format. For those cases, the output value of **pl_tadd** cannot be converted to external format (it should lead to an error).

7.9.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **pl_tadd** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the PPF_LIB software library **pl_vector_msg** (see RD 3).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value.These error codes can be obtained translating the extended status flag returned by the **pl_tadd** function by calling the function of the PPF_LIB software library **pl_vector_code** (see RD 3)

Error type	Error message	Cause and impact	Error code	Error No
ERR	Input NUMBER of DAYS is out of limits (-18262,36524)	No calculation performed	PL_CFI_K22_NDAYS_ERR	0
ERR	Input SECONDS is out of limits (0-86399)	No calculation performed	PL_CFI_K22_NSECS_NDAYS_ERR	1
ERR	Input MICROSECONDS is out of limits (0-999999)	No calculation performed	PL_CFI_K22_NMICS_ERR	2
ERR	Input DELTA UT1 is out of limits (+-999999)	No calculation performed	PL_CFI_K22_NDUT1_ERR	3
WARN	Final NUMBER of DAYS is out of limits (-18262,36524)	Not allowed value	PL_CFI_K22_NDAYS_WARN	4
WARN	Final DELTA UT1 is out of limits (+-999999)	Not allowed value	PL_CFI_K22_NDUT1_WARN	7

7.9.6 Runtime performances

The following runtime performances have been measured.

Ultra Sparc [ms]
0.029

7.10 pl_tsub

7.10.1 Overview

The **pl_tsub** CFI function substracts two UT1 time expressed in Transport format

7.10.2 Calling interface

The calling interface of the **pl_tsub** CFI function is the following (input parameters are underlined):

```
#include <ppf_lib.h>
{
      long mjdt1[4], mjdt2[4], subt[4], status;

      status = pl_tsub(mjdt1, mjdt2, subt);
}
```

For Fortran programs the declaration and calling procedure is as follows (input parameters are underlined, note that the C preprocessor must be used because of the presence of the `#include` statement):

```
#include <ppf_lib.inc>

      INTEGER*4 MJDT1(4), MJDT2(4), SUBT(4), STATUS

      STATUS = PL_TSUB (MJDT1, MJDT2, SUBT)
```

7.10.3 Input parameters

The **pl_tsub** CFI function has the following input parameters:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
mjdt1[4]	long	[0]	UTC of the first date (UT1 time)	integer days (Transport format)	>= -18262 <= +36524
		[1]		integer seconds (Transport format)	>= 0 < 86400
		[2]		integer μs (Transport format)	>= 0 <= 999999
		[3]	ΔUT1 of the first date (UT1 time)	integer μs (Transport format)	>= -999999 <= +999999
mjdt2[4]	long	[0]	UTC of the second date (UT1 time)	integer days (Transport format)	>= -18262 <= +36524
		[1]		integer seconds (Transport format)	>= 0 < 86400
		[2]		integer μs (Transport format)	>= 0 <= 999999
		[3]	ΔUT1 of the second date (UT1 time) Important note: unless exceptional circumstances ΔUT1 values should not change within a product, thus this parameter should be identical to mjdt1[3]	integer μs (Transport format)	>= -999999 <= +999999

7.10.4 Output parameters

The output parameters of the **pl_tsub** CFI function are:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
pl_tsub	long	-	Extended status flag	-	-
subt[4]	long	[0]	Resulting UTC difference mjdt1 - mjdt2 (UT1 time)	integer days (Transport format)	>= -18262 <= +36524
		[1]		integer seconds (Transport format)	>= 0 < 86400
		[2]		integer μs (Transport format)	>= 0 <= 999999
		[3]	Resulting ΔUT1 difference mjdt2 - mjdt1 (UT1 time)	integer μs (Transport format)	>= -1999999 <= +1999999

Note that, for some cases, the resulting ΔUT1 may exceed the maximum value of 0.999999 seconds allowed for the external format. For those cases, the output value of **pl_tsub** cannot be converted to external format (it should lead to an error).

7.10.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **pl_tsub** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the PPF_LIB software library **pl_vector_msg** (see RD 3).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the extended status flag returned by the **pl_tsub** function by calling the function of the PPF_LIB software library **pl_vector_code** (see RD 3)

Error type	Error message	Cause and impact	Error code	Error No
ERR	Input NUMBER of DAYS is out of limits (-1862,36524)	No calculation performed	PL_CFI_K23_NDAYS_ERR	0
ERR	Input SECONDS is out of limits (0-86399)	No calculation performed	PL_CFI_K23_NSECS_NDAYS_ERR	1
ERR	Input MICROSECONDS is out of limits (0-999999)	No calculation performed	PL_CFI_K23_NMICS_ERR	2
ERR	Input DELTA UT1 is out of limits (+-999999)	No calculation performed	PL_CFI_K23_NDUT1_ERR	3
WARN	Final NUMBER of DAYS is out of limits (-1862,36524)	Not allowed value	PL_CFI_K23_NDAYS_WARN	4
WARN	Final DELTA UT1 is out of limits (+-1999999)	Not allowed value	PL_CFI_K23_NDUT1_WARN	7

7.10.6 Runtime performances

The following runtime performances have been measured.

Ultra Sparc [ms]
0.029

7.11 pl_sbtutc

7.11.1 Overview

The **pl_sbtutc** CFI function transforms from OBT count to UTC time. The name **pl_sbtutc** is historical.

See RD 2 for details on time formats and representations, in particular the definitions of SBT and OBT.

Note that there is an ambiguity on the UTC to be computed, because a given OBT count corresponds to many possible UTC times. This is due to the wrap-around of the OBT counter, which occurs about every 190 days.

To solve the ambiguity, the chosen UTC time (given as output) is the UTC time nearest to the reference UTC (given as input) and corresponding to the specified OBT (also given as input).

7.11.2 Calling interface

The calling interface of the **pl_sbtutc** CFI function is the following (input parameters are underlined):

```
#include <ppf_lib.h>
{
      long mjdt0[4], mjdt[4], status;
      unsigned long sbt0, per0, obtm, obtl;

      status = pl_sbtutc (mjdt0, &sbt0, &per0, &obtm, &obtl, mjdt);
}
```

This function cannot be called from Fortran (see 8)

7.11.3 Input parameters

The **pl_sbtutc** CFI function has the following input parameters:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
mjdt0[4]	long	[0]	Reference UTC (UTC time)	integer days (Transport format)	>= -18262 <= +36524
		[1]		integer seconds (Transport format)	>= 0 < 86400
		[2]		integer μs (Transport format)	>= 0 <= 999999
		[3]	DUMMY		
sbt0	unsigned long *	-	Counter containing the SBT at the reference UTC	256 Hz	>= 0x00000000 <= 0xFFFFFFFF
per0	unsigned long *	-	Actual Envisat-1 on-board clock period (aprox 256 Hz)	picoseconds [10^{-12} s]	> 0x00000000 <= 0xFFFFFFFF
obtm	unsigned long *	-	Counter containing the specified OBT 32 most significant bits	counter [256 Hz]	>= 0x00000000 <= 0xFFFFFFFF
obtl	unsigned long *	-	Counter containing the specified OBT 32 least significant bits, right-padded with zeros if needed	counter [65536 Hz]	>= 0x00000000 <= 0xFFFFFFFF

7.11.4 Output parameters

The output parameters of the **pl_sbtutc** CFI function are:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
pl_sbtutc	long	-	Extended status flag	-	-
mjdt[4]	long	[0]	UTC nearest to mjdt0 corresponding to specified OBT. (UTC time)	integer days (Transport format)	>= -18262 <= +36524
		[1]		integer seconds (Transport format)	>= 0 < 86400
		[2]		integer μs (Transport format)	>= 0 <= 999999
		[3]	DUMMY (i.e. not overridden)		

7.11.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **pl_sbtutc** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the PPF_LIB software library **pl_vector_msg** (see RD 3).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the extended status flag returned by the **pl_sbtutc** function by calling the function of the PPF_LIB software library **pl_vector_code** (see RD 3)

Error type	Error message	Cause and impact	Error code	Error No
ERR	Input NUMBER of DAYS is out of limits (-18262,36524)	No calculation performed	PL_CFI_K24_NDAYS_ERR	0
ERR	Input SECONDS is out of limits (0-86399)	No calculation performed	PL_CFI_K24_NSECS_NDAYS_ERR	1
ERR	Input MICROSECONDS is out of limits (0-999999)	No calculation performed	PL_CFI_K24_NMICS_ERR	2
ERR	Input DELTA UT1 is out of limits (+-999999)	No calculation performed	PL_K24_NDUT1_ERR	3
ERR	Input SBT PERIOD is null	No calculation performed	PL_K24_PER_ERR	4

7.11.6 Runtime performances

The following runtime performances have been measured.

Ultra Sparc [ms]
0.035

7.12 pl_utcsbt

7.12.1 Overview

The **pl_utcsbt** CFI function transforms from UTC time to OBT count. The name **pl_utcsbt** is historical. See RD 2 for details on time formats and representations, in particular the definitions of SBT and OBT.

Note that no rounding to any number of significant bits is performed by **pl_utcsbt**. The user application must perform this rounding if necessary. An example of rounding is provided in the example program delivered with the PPF_LIB library.

7.12.2 Calling interface

The calling interface of the **pl_utcsbt** CFI function is the following (input parameters are underlined):

```
#include <ppf_lib.h>
{
      long mjdt0[4], mjdt[4], status;
      unsigned long sbt0, per0, obtm, obtl;

      status = pl_utcsbt (mjdt0, &sbt0, &per0, &obtm, &obtl, mjdt);
}
```

This function cannot be called from Fortran (see 8)

7.12.3 Input parameters

The **pl_utcsbt** CFI function has the following input parameters:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
mjdt0[4]	long	[0]	Reference UTC (UTC time)	integer days (Transport format)	>= -18262 <= +36524
		[1]		integer seconds (Transport format)	>= 0 < 86400
		[2]		integer μs (Transport format)	>= 0 <= 999999
		[3]	DUMMY		
sbt0	unsigned long *	-	Counter containing the SBT at the reference UTC	256 Hz	>= 0x00000000 <= 0xFFFFFFFF
per0	unsigned long *	-	Actual Envisat-1 on-board clock period (approx 256 Hz)	picoseconds [10^{-12} s]	> 0x00000000 <= 0xFFFFFFFF
mjdt[4]	long	[0]	Specified UTC (UTC time)	integer days (Transport format)	>= -18262 <= +36524
		[1]		integer seconds (Transport format)	>= 0 < 86400
		[2]		integer μs (Transport format)	>= 0 <= 999999
		[3]	DUMMY		

7.12.4 Output parameters

The output parameters of the **pl_utcsbt** CFI function are:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
pl_utcsbt	long	-	Extended status flag	-	-
obtm	unsigned long *	-	Counter containing the OBT 32 most significant bits at the specified UTC	counter [256 Hz]	>= 0x00000000 <= 0xFFFFFFFF
obtl	unsigned long *	-	Counter containing the OBT 32 least significant bits at the specified UTC	counter [65536 Hz]	>= 0x00000000 <= 0xFFFFFFFF

7.12.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **pl_utcsbt** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the PPF_LIB software library **pl_vector_msg** (see RD 3).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the extended status flag returned by the **pl_utcsbt** function by calling the function of the PPF_LIB software library **pl_vector_code** (see RD 3)

Error type	Error message	Cause and impact	Error code	Error No
ERR	Input NUMBER of DAYS is out of limits (-18262,36524)	No calculation performed	PL_CFI_K25_NDAYS_ERR	0
ERR	Input SECONDS is out of limits (0-86399)	No calculation performed	PL_CFI_K25_NSECS_ERR	1
ERR	Input MICROSECONDS is out of limits (0-999999)	No calculation performed	PL_CFI_K25_NMICS_ERR	2
ERR	Input DELTA UT1 is out of limits (+-999999)	No calculation performed	PL_K25_NUTC1_ERR	3
ERR	Input SBT PERIOD is null	No calculation performed	PL_K25_PER_ERR	4

7.12.6 Runtime performances

The following runtime performances have been measured.

Ultra Sparc [ms]
0.033

7.13 pl_sun

7.13.1 Overview

The **pl_sun** CFI function calculates the position and velocity vector of the Sun in the True of Date coordinate system

7.13.2 Calling interface

The calling interface of the **pl_sun** function is the following (input parameters are underlined):

```
#include <ppf_lib.h>
{
      double utc[2], rsun[3], rdsun[3];
      long status;

      status = pl_sun(utc, rsun, rdsun);
}
```

For Fortran programs the declaration and calling procedure is as follows (input parameters are underlined, note that the C preprocessor must be used because of the presence of the `#include` statement):

```
#include <ppf_lib.inc>

      REAL*8 UTC(2), RSUN(3), RDSUN(3)
      INTEGER*4 STATUS

      STATUS = PL_SUN(UTC, RSUN, RDSUN)
```

7.13.3 Input parameters

The **pl_sun** CFI function has the following input parameters:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
utc[2]	double	[0]	Specified time (UTC time)	decimal days (MJD2000)	-
		[1]	DUMMY	-	-

7.13.4 Output parameters

The output parameters of the **pl_sun** CFI function are:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
pl_sun	long	-	Extended status flag	-	-
rsun[3]	double	all	Position vector of the Sun in the True of Date CS	m	-
rdsun[3]	double	all	Velocity vector of the Sun in the True of Date CS	m/s	-

7.13.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **pl_sun** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the PPF_LIB software library **pl_vector_msg** (see RD 3).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the extended status flag returned by the **pl_sun** function by calling the function of the PPF_LIB software library **pl_vector_code** (see RD 3)

Error type	Error message	Cause and impact	Error code	Error No
	(no errors)			

7.13.6 Runtime performances

The following runtime performances have been measured.

Ultra Sparc [ms]
0.715

7.14 pl_moon

7.14.1 Overview

The **pl_moon** CFI function calculates the position and velocity vector of the Moon in the True of Date co-ordinate system

7.14.2 Calling interface

The calling interface of the **pl_moon** CFI function is the following (input parameters are underlined):

```
#include <ppf_lib.h>
{
       double utc[2], rmoon[3], rdmoon[3];
       long status;

       status = pl_moon(utc, rmoon, rdmoon);
}
```

For Fortran programs the declaration and calling procedure is as follows (input parameters are underlined, note that the C preprocessor must be used because of the presence of the `#include` statement):

```
#include <ppf_lib.inc>

       REAL*8 UTC(2), RMOON(3), RDMOON(3)
       INTEGER*4 STATUS

       STATUS = PL_MOON(UTC, RMOON, RDMOON)
```

7.14.3 Input parameters

The **pl_moon** CFI function has the following input parameters:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
utc[2]	double	[0]	Specified time (UTC time)	decimal days (MJD 2000)	-
		[1]	DUMMY	-	-

7.14.4 Output parameters

The output parameters of the **pl_moon** CFI function are:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
pl_moon	long	-	Extended status flag	-	-
rmoon[3]	double	all	Position vector of the Moon in the True of Date CS	m	-
rdmoon[3]	double	all	Velocity vector of the Moon in the True of Date CS	m/s	-

7.14.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **pl_moon** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the PPF_LIB software library **pl_vector_msg** (see RD 3).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the extended status flag returned by the **pl_moon** function by calling the function of the PPF_LIB software library **pl_vector_code** (see RD 3)

Error type	Error message	Cause and impact	Error code	Error No
	(no errors)			

7.14.6 Runtime performances

The following runtime performances have been measured.

Ultra Sparc [ms]
0.794

7.15 pl_planets

7.15.1 Overview

The **pl_planets** CFI function calculates the position and velocity vector of a planet in the Heliocentric Mean of 2000.0 coordinate system

7.15.2 Calling interface

The calling interface of the **pl_planets** CFI function is the following (input parameters are underlined):

```
#include <ppf_lib.h>
{
       long planet;
       double utc[2], rplanet[3], rdplanet[3];
       long status;

       status = pl_planets(&planet, utc, rplanet, rdplanet);
}
```

For Fortran programs the declaration and calling procedure is as follows (input parameters are underlined, note that the C preprocessor must be used because of the presence of the `#include` statement):

```
#include <ppf_lib.inc>

       INTEGER*4 PLANET
       REAL*8 UTC(2), RPLANET(3), RDPLANET(3)
       INTEGER*4 STATUS

       STATUS = PL_PLANETS(PLANET, UTC, RPLANET, RDPLANET)
```

7.15.3 Input parameters

The **pl_planets** CFI function has the following input parameters:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
planet	long *	-	Planet ID	-	1 - 8
utc[2]	double	[0]	Specified time (UTC time)	decimal days (MJD2000)	-
		[1]	DUMMY	-	-

It is possible to use enumeration values rather than integer values for some of the input arguments:

Input	Description	Enumeration value	Long
planet	Mercury ID	PL_MERCURY	1
	Venus ID	PL_VENUS	2
	Earth-Moon barycenter ID	PL_EM_BAR	3
	Mars ID	PL_MARS	4
	Jupiter ID	PL_JUPITER	5
	Saturn ID	PL_SATURN	6
	Uranus ID	PL_URANUS	7
	Neptune ID	PL_NEPTUNE	8

7.15.4 Output parameters

The output parameters of the **pl_planets** CFI function are:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
pl_planets	long	-	Extended status flag	-	-
rplanet[3]	double	all	Position vector of the Planet in the Heliocentric Mean of 2000.0 coordinate system	m	-
rdplanet[3]	double	all	Velocity vector of the Planet in the Heliocentric Mean of 2000.0 coordinate system	m/s	-

7.15.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **pl_planets** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the PPF_LIB software library **pl_vector_msg** (see RD 3).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the extended status flag returned by the **pl_planets** function by calling the function of the PPF_LIB software library **pl_vector_code** (see RD 3)

Error type	Error message	Cause and impact	Error code	Error No
ERR	Wrong selected planet on input (not allowed) (Input PLANET ID Code is out of limits(1-8))	Calculation not performed	PL_CFI_K04_PLA_ERR	2

7.15.6 Runtime performances

The following runtime performances have been measured.

Two runtimes are provided, one with fixed inputs, i.e. the functions has been called several times with the same UT1 time; and a second one with random inputs (random UT1 time).

Ultra Sparc [ms] RANDOM inputs	Ultra Sparc [ms] FIXED inputs
0.121	0.100

7.16 pl_star_radec

7.16.1 Overview

The **pl_star_radec** CFI function calculates the right ascension and declination of a star in the True of Date coordinate system.

7.16.2 Calling interface

The calling interface of the **pl_star_radec** CFI function is the following (input parameters are underlined):

```
#include <ppf_lib.h>
{
       double utc[2], ra0, dec0, mu_ra, mu_dec;
       double rad_vel, par, ra, dec;
       long status;

       status = pl_star_radec(utc, &ra0, &dec0,
                              &mu_ra, &mu_dec, &rad_vel, &par,
                              &ra, &dec);
}
```

For Fortran programs the declaration and calling procedure is as follows (input parameters are underlined, note that the C preprocessor must be used because of the presence of the `#include` statement):

```
#include <ppf_lib.inc>

        REAL*8 UTC(2),RA0,DEC0,MU_RA,MU_DEC
        REAL*8 RAD_VEL,PAR,RA,DEC
        INTEGER*4 STATUS

        STATUS = PL_STAR_RADEC(UTC, RA0, DEC0,
       &                       MU_RA, MU_DEC, RAD_VEL, PAR
       &                       RA,DEC)
```

7.16.3 Input parameters

The **pl_star_radec** CFI function has the following input parameters:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
utc[2]	double	[0]	Specified time (UTC time)	decimal days (MJD2000)	-
		[1]	DUMMY	-	-
ra0	double *	-	Right ascension of the star at J2000.0 (Barycentric Mean of 2000.0 CS)	rad	>= 0 < 2π
dec0	double *	-	Declination of the star at J2000.0 (Barycentric Mean of 2000.0 CS)	rad	>= -π/2 <= +π/2
mu_ra	double *	-	Proper motion in the right ascension at J2000.0 (Barycentric Mean of 2000.0 CS)	rad/century	-
mu_dec	double *	-	Proper motion in the declination at J2000.0 (Barycentric Mean of 2000.0 CS)	rad/century	-
rad_vel	double *	-	Radial velocity of the star at J2000.0 (Barycentric Mean of 2000.0 CS)	au/century	-
par	double *	-	Parallax of the star at J2000.0 (Barycentric Mean of 2000.0 CS)	rad	-

7.16.4 Output parameters

The output parameters of the **pl_star_radec** CFI function are:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
pl_star_radec	long	-	Extended status flag	-	-
ra	double *	-	Right ascension of the star at specified time (True of Date CS)	rad	>= 0 < 2π
dec	double *	-	Declination of the star at specified time (True of Date CS)	rad	>= -π/2 <= +π/2

Note that the output right ascension and declination are not in degrees.

7.16.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **pl_star_radec** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the PPF_LIB software library **pl_vector_msg** (see RD 3).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation.

The table is completed by the error code and value. These error codes can be obtained translating the extended status flag returned by the **pl_star_radec** function by calling the function of the PPF_LIB software library **pl_vector_code** (see RD 3)

Error type	Error message	Cause and impact	Error code	Error No
	(no errors)			

The declination is not checked, so in case it does not satisfy its allowed range it may result in raising an internal error (see 8).

7.16.6 Runtime performances

The following runtime performances have been measured.

Two runtimes are provided, one with fixed inputs, i.e. the functions has been called several times with the same UT1 time, but modifying the other input parameters; and a second one with random inputs, i.e all the inputs have been modified from call to call and the average time has been taken.

Ultra Sparc [ms] RANDOM inputs	Ultra Sparc [ms] FIXED inputs
0.723	0.054

8 LIBRARY PRECAUTIONS

The following precaution shall be taking into account when using PPF_LIB library:

- The functions *pl_sbtutc* and *pl_utcsbt* cannot be called from Fortran (no Fortran equivalence to *unsigned long*).
- When a message like

PPF_LIB >>> ERROR in *pl_function*: Internal computation error # *n*
or
PPF_LIB >>> WARNING in *pl_function*: Internal computation warning # *n*

appears, run the program in *verbose* mode for a complete description of warnings and errors and call for maintenance if necessary

.

9 KNOWN PROBLEMS

The following precautions shall be taken into account when using the CFI software libraries:

CFI library	Problem	Work around solution
pl_sbtutc	Cannot be called from Fortran (no Fortran equivalence to unsigned long)	-
pl_utcsbt	Cannot be called from Fortran (no Fortran equivalence to unsigned long)	-