Earth Observation
Mission CFI Software

EO_LIB
SOFTWARE USER MANUAL

Code:	EO-MA-DMS-GS-0003
Issue:	4.14
Date:	16/11/2017

	Name	Function	Signature
Prepared by:	Mariano Sánchez-Nogales	Project Engineer	
	Fabrizio Pirondini	Project Engineer	
	Juan Jose Borrego Bote	Project Engineer	
	Carlos Villanueva	Project Engineer	
	Rubén Castro	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:	15583/01/NL/GS	Internal	
		Public	
Contract Issuer:	ESA / ESTEC	Industry	X
		Confidential	

External Distribution		
Name	Organisation	Copies

Electronic handling	
Word Processor:	LibreOffice 5.2.3.3
Archive Code:	P/SUM/DMS/01/026-008
Electronic file name:	eo-ma-dms-gs-003-21

DOCUMENT STATUS LOG

Issue	Change Description	Date	Approval
1.0	New document	08/11/01	
1.1	Updated Time Transformation functions	04/02/02	
1.2	Updated the following functions: xl_change_cart_cs, xl_geod_to_cart, xl_cart_to_geod, xl_kepl_to_cart, xl_cart_to_kepl, xl_sun, xl_moon, xl_planet, xl_star_radec, xl_geod_distance, xl_time_ref_init_file, xl_time_ref_close. The xl_attitude_cs function has been removed and replaced by xp_attitude in the EXPLORER_POINTING library.	15/04/02	
1.3	Added xl_time_get_leap_second_info	19/07/02	
2.0	Maintenance release.	29/11/02	
2.1	Maintenance release.	13/05/03	
2.2	Added xl_default_sat_init function.	30/09/03	
3.0	New initialisation strategy and interfaces	21/07/04	
3.1	Maintenance Release. New functions: - xl_get_rotation_angles, - xl_get_rotated_vectors, - xl_position_on_orbit	13/10/04	
3.2	Maintenance release.	15/11/04	
3.3	Maintenance release. New features: - Changes for dealing with the new library explorer_data_handling - Identifier accessors. - OBT to UTC conversion for ADM and SMOS - Support for ENVISAT ASCII files removed	11/07/05	
3.4	Maintenance release. New function xl_default_sat_close.	18/11/05	

3.5	Maintenance release. New features for xl_time_ref_init_file. New features for xl_change_cart_cs New functions prototypes: - xl_cart_to_radec - xl_radec_to_cart - xl_star_catalog - xl_topocentric_to_ef - xl_ef_to_topocentric	26/05/06	
3.6	Maintenance release. New features: - xl_change_cart_cs - SMOS UTC proteus time format - Parameters for SENTINEL-1 New functions implemented: - xl_euler_to_matrix and xl_matrix_to_euler - xl_cart_to_radec and xl_radec_to_cart - xl_star_catalog - xl_topocentric_to_ef and xl_ef_to_topocentric	24/11/06	
3.7	Maintenance release. New features: - Function expcfi_check_libs - Library version for MAC OS X on Intel (32 and 64-bits)	13/07/07	
3.8	Maintenance release. New features: - Parameters for SENTINEL-2, SENTINEL-3 and SEOSAT - Generic Satellite	31/07/08	
4.0	Maintenance release. New features: - Function interfaces changed for model support	19/01/09	
4,1	Maintenance release. New features: - Time initialization with list of files - Time initialization with OSF	07/05/10	
4.2	Maintenance release.	31/01/11	

4.3	Maintenance release. New features: - Pseudo-EF CS added. - Polar motion included in EF CS. - New init function xl_time_id_init. - TimeId initialization support IERS Bulletin A and Bulletins A+B - New time transport formats: - XL_TRANS_GENERIC_GPS - XL_TRANS_GENERIC_GPS_WEEK		
4.4	Maintenance release. New features: – New non-iterative method to compute transformation from cartesian to geodetic coordinates. – New reference frames for on-board position scheduling: EF and GM2000		
4.5	Maintenance release: New features: - New function xl_geoid_calc, to transform between heights relative to the ellipsoid and the geoid		
4.6	Maintenance release.		
4.7	Support for SENTINEL_5P, JASON-CS AND METOP-ST satellites.	03/28/14	
4.8	Maintenance release: New features: - New reference frame: Earth Fixed non rotating (intermediate step to the Greenwich reference frame) - New Sun model to take into account Sun light travel time. - New function for quaternions interpolation: xl_quaternions_interpol	29/10/2014	

4.9	Maintenance release	23/04/2015	
4.10	Maintenance release	29/10/2015	
4.11	Maintenance release New features: - Support for BIOMASS, SENTINEL-5 and SAOCOM-CS satellites	15/04/2016	
4.12	Maintenance release New features: - Extrapolation algorithm implemented for quaternions (xl_quaternions_intepol)	03/11/2016	
4.13	Maintenance release	05/04/2017	
4.14	Maintenance release New features: - New functions for CUC time managing: xl_time_cuc_to_processing xl_time_processing_to_cuc - Support for FLEX satellite	16/11/2017	

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 SCOPE

The EO_LIB Software User Manual provides a detailed description of usage of the CFI functions included within the EO_LIB CFI software library.

2 ACRONYMS, NOMENCLATURE AND TERMINOLOGY

2.1 Acronyms

ANX	Ascending Node Crossing
AOCS	Attitude and Orbit Control Subsystem
ASCII	American Standard Code for Information Interchange
BOM	Beginning Of Mission
CFI	Customer Furnished Item
EGM96	Earth Gravitational Model 1996
EO	Earth Observation
EOM	End Of Mission
ESA	European Space Agency
ESTEC	European Space Technology and Research Centre
GPL	GNU Public License
GPS	Global Positioning System
IERS	International Earth Rotation Service
I/F	Interface
LS	Leap Second
OBT	On-board Binary Time
OSF	Orbit Scenario File
SRAR	Satellite Relative Actual Reference
SUM	Software User Manual
TAI	International Atomic Time
UTC	Coordinated Universal Time
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)

2.3 Note on Terminology

In order to keep compatibility with legacy CFI libraries, the Earth Observation Mission CFI Software makes use of terms that are linked with missions already or soon in the operational phase like the Earth Explorers.

This may be reflected in the rest of the document when examples of Mission CFI Software usage are proposed or description of Mission Files is given.

3 APPLICABLE AND REFERENCE DOCUMENTS

3.1 Applicable Documents

No applicable documents.

3.2 Reference Documents

[MCD]	Earth Observation Mission CFI Software. Conventions Document. EO-MA-DMS-GS-0001.
[GEN_SUM]	Earth Observation Mission CFI Software. General Software User Manual. EO-MA- DMS-GS-0002.
[F_H_SUM]	Earth Observation Mission CFI Software. EO_FILE_HANDLING Software User Manual. EO-MA-DMS-GS-0008.
[D_H_SUM]	Earth Observation Mission CFI Software. EO_DATA_HANDLING Software User Manual. EO-MA-DMS-GS-007.
[IERS]	http://www.iers.org/iers/publications/bulletins/
[CUC]	CCSDS TIME CODE FORMATS RECOMMENDED STANDARD, CCSDS 301.0-B, section 3.2

The latest applicable version of [MCD], [GEN_SUM], [F_H_SUM], [DH_SUM] is v4.14 and can be found at: http://eop-cfi.esa.int/REPO/PUBLIC/DOCUMENTATION/CFI/EOCFI/BRANCH_4X/

4 INTRODUCTION

4.1 Functions Overview

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

The following CFI functions are included:

4.1.1 Time Computations

All time time computations are performed internally using the continuous TAI time reference. Therefore the input and output parameters are converted internally to the adequate time reference.

4.1.1.1 Time Reference Transformations Initialization

- **xl_time_ref_init_file**: initializes time correlations between TAI, UTC, UT1 and GPS times from reference data files.
- **xl_time_ref_init**: initializes time correlations between TAI, UTC, UT1 and GPS times from input reference times.
- **xl_time_close**: cleans up any memory allocation performed by the initialization functions.
- **xl_time_get_leap_second_info**: retrieves the leap second location (if any) in the initialised time range.

4.1.1.2 Time Format and Reference Transformations

- **xl_time_ascii_to_ascii**: transforms a time expressed in a given ASCII format and reference (TAI, UTC, UT1 or GPS) into a time in a different ASCII format and/or reference (TAI, UTC, UT1 or GPS).
- **xl_time_ascii_to_transport**: transforms a time expressed in a given ASCII format and reference (TAI, UTC, UT1 or GPS) into a time in a Transport format, performing a reference transformation if necessary (to TAI, UTC, UT1 or GPS).
- **xl_time_ascii_to_processing**: transforms a time expressed in a given ASCII format and reference (TAI, UTC, UT1 or GPS) into a time in Processing format, performing a reference transformation if necessary (to TAI, UTC, UT1 or GPS).
- **xl_time_processing_to_ascii**: transforms a time expressed in Processing format and a given reference (TAI, UTC, UT1 or GPS) into a time in an ASCII format, performing a reference transformation if necessary (to TAI, UTC, UT1 or GPS).
- **xl_time_processing_to_transport**: transforms a time expressed in Processing format and a given reference (TAI, UTC, UT1 or GPS) into a time in a Transport format, performing a reference transformation if necessary (to TAI, UTC, UT1 or GPS).
- **xl_time_processing_to_processing**: transforms a time expressed in Processing format and a given reference (TAI, UTC, UT1 or GPS) into a time in Processing format with a different reference (TAI, UTC, UT1 or GPS).
- **xl_time_transport_to_ascii**: transforms a time expressed in a given Transport format and reference (TAI, UTC, UT1 or GPS) into a time in an ASCII format, performing a reference transformation if necessary (to TAI, UTC, UT1 or GPS).

- **xl_time_transport_to_transport**: transforms a time expressed in a given Transport format and reference (TAI, UTC, UT1 or GPS) into a time in a different Transport format and/or reference (TAI, UTC, UT1 or GPS).
- **xl_time_transport_to_processing**: transforms a time expressed in a given Transport format and reference (TAI, UTC, UT1 or GPS) into a time in Processing format, performing a reference transformation if necessary (to TAI, UTC, UT1 or GPS).

4.1.1.3 Operation between Dates

- **xl_time_add**: adds a duration to a TAI, UTC, UT1 or GPS time expressed in Processing format.
- **xl_time_diff**: subtracts two TAI, UTC, UT1 or GPS times expressed in Processing format.

4.1.1.4 Transformations from/to On-board Times

- **xl_time_obt_to_time**: transforms an On-board Time (OBT) into a TAI, UTC, UT1 or GPS time in Processing format.
- **xl_time_time_to_obt**: transforms a TAI, UTC, UT1 or GPS time expressed in Processing format into an On-board Time (OBT).

4.1.2 Coordinate Systems Transformations

4.1.2.1 Reference Frames Transformations

- **xl_change_cart_cs**: transforms a state vector between different coordinate systems.
- **xl_topocentric_to_ef**: transforms a state vector from topocentric coordinates to the Earth Fixed CS.
- **xl_ef_to_topocentric**: transforms a state vector from the Earth Fixed CS to topocentric coordinates.

4.1.2.2 Attitude-related Computations

- **xl_euler_to_matrix**: computes the elements of the coordinate transformation matrix with respect to the attitude frame given the corresponding Euler rotation vector in the roll, pitch and yaw sequence.
- **xl_matrix_to_euler**: derives the Euler rotation vector with respect to the attitude frame in the roll, pitch and yaw sequence given the corresponding coordinate transformation matrix.
- **xl_get_rotation_angles**: calculates the rotation angles between two sets of orthonormal right-handed unit vectors expressed wrt an identical coordinate frame.
- **xl_get_rotated_vectors**: calculates the rotated unit vectors given a set of unit vectors and the rotation angles expressed wrt an identical coordinate frame.
- **xl_quaternions_to_vectors**: calculates the orthonormal unit vectors from a given set of quaternions.
- **xl_vectors_to_quaternions**: calculates the set of quaternions that correspond to a set of orthonormal unit vectors.

4.1.2.3 Coordinates Transformations

- **xl_geod_to_cart**: transforms from Geodetic to Cartesian coordinates.
- **xl_cart_to_geod**: transforms from Cartesian to Geodetic coordinates.
- **xl_cart_to_radec**: transforms from a cartesian vector to right ascension and declination.
- **xl_radec_to_cart**: transforms from right ascension and declination to a cartesian vector.

4.1.2.4 State Vector Transformations

- **xl_kepl_to_cart**: transforms from Keplerian to Cartesian coordinates.
- **xl_cart_to_kepl**: transforms from Cartesian to Keplerian coordinates.

4.1.2.5 Position on orbit calculations

- **xl_position_on_orbit**: calculates a value describing the position of the satellite within the orbit, using as input a Cartesian orbit state vector.

4.1.2.6 Quaternions transformations

- **xl_quaternions_interpol**: interpolates a quaternion using the spherical linear interpolation method.

4.1.3 Other Basic Computations

- **xl_sun**: calculates the position and velocity of the Sun in the Earth Fixed coordinate system
- **xl_moon**: calculates the Moon position and velocity in the Earth Fixed coordinate system
- **xl_planet**: calculates the position and velocity of a selected planet in the Earth Fixed coordinate system
- **xl_star_radec**: calculates the right ascension and declination of a star in the True of Date coordinate system.
- **xl_geod_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.
- **xl_star_catalog**: calculates the star coordinates in a star catalogue reference frame.

4.1.4 Astronomical model selection

- **xl_model_init**: It initialises a model identifier that will be used to by other CFI functions to select a model.
- **xl_model_close**: cleans up any memory allocation performed by the initialization functions.

4.2 Time Reference Transformations Calling Sequence

Time reference transformations ,and other functions with time as input, requires the user to initialise correlations between the different allowed time references, i.e. TAI, UTC, UT1 and GPS time. In order to accomplish such correlations, two possible strategies can be used:

- Initialization from a single or multiple orbit files (**xl_time_ref_init_file**)
- Initializationwith data structures (user data or data read from files)(**xl_time_id_data**).
- Initialization from a given set of time references (**xl_time_ref_init**).

The correlations are stored in a data structure, and the software returns a pointer to it, in addition to the validity range of the initialisation. This structure is referred to as the *Time Id*.

Once the initialisation has been performed, the user is able to transform any date expressed in one of the allowed time references to another, through the Time Format / Reference Transformation functions. The *Time ID* has to be provided to each of these functions. The process can be repeated as needed without initialising the time correlations each time.

After finalising the transformations, the *Time ID* must be freed (**xl_time_close**).

A complete view of the time reference transformations sequence is presented in figure 1.

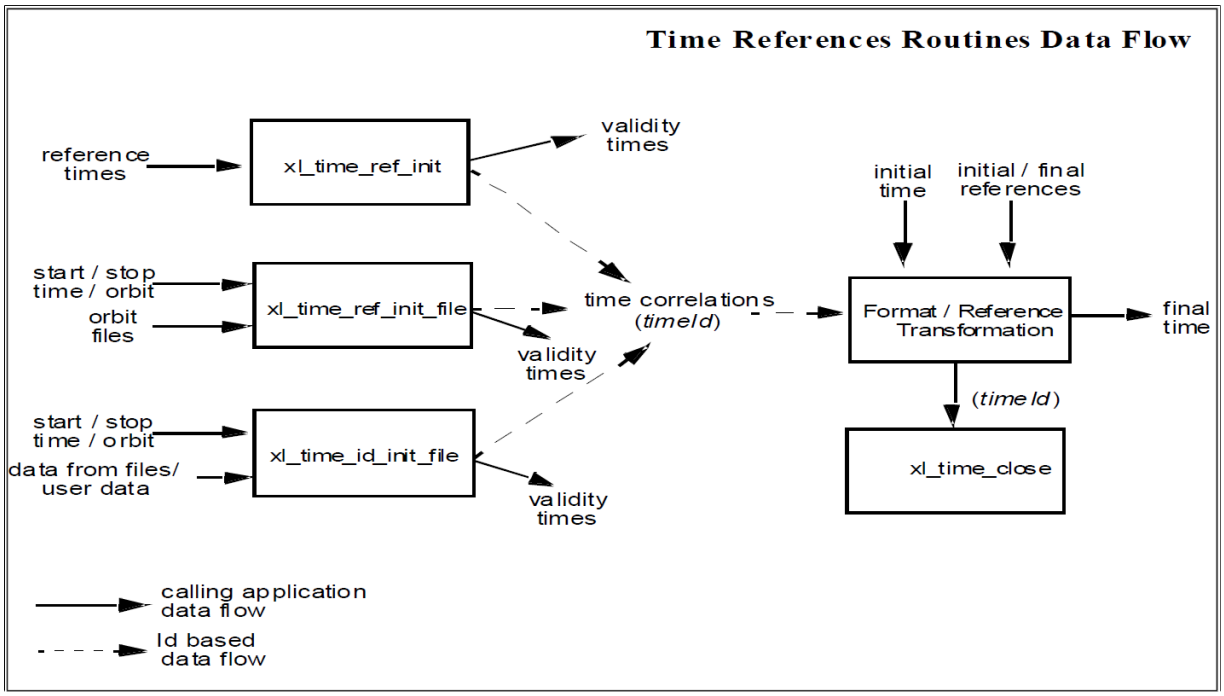


Figure 1: Time reference transformations sequence

There is a second way of calling the functions that require a *timeId* as input.

Similar initialisation functions exist in other CFI libraries, resulting in various *Ids* being generated. It is possible to group different *Ids* into a single entity called *runId*. Using this method, a single *runId* can be passed to all functions across the different libraries, instead of passing several *Ids* through the interface.

In this case, the first step would be to create the *timeId*. Then, a *runId* can be generated using as input the *timeId*. This *runId* is then passed through the interface to equivalent functions to those described before (ending in "_run").

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

Please refer also to:

- [MCD] for a detailed description of the time references and formats, coordinate systems, parameters and models used in this document.
- [GEN_SUM] for a complete overview of the CFI, and in particular the detailed description of the Id concept and the error handling functions.

4.3 Earth and Astronomical model selection calling sequence

The CFI functions can work with different Earth and astronomical models. These models have been divided in the following categories:

- Star model
- Sun model
- Planet model
- Earth model
- Moon model

- Nutation model
- Precession model
- Constants model

In order to work with different models, these have to be stored in a CFI Id called *Model ID*. The *Model ID* is a variable of type **xl_model_id**.

The calling sequence for a C program where the *Model ID* is needed, would be as follows:

- Declare the model_id variable:
  - **`xl_model_id`** `model_id = {NULL};`
  - The model_id has to be initialised this way (as other CFI ID's), so that the EOCFI could recognise that the model_id is not initialised.
- The user is required to explicitly initialize the model_id with the **xl_model_init** function (see section 7.51). In case this is not done and a model_id set to {NULL} is passed to a CFI function, that CFI function will allocate and use a temporary model_id (set to default models) that will be released at function completion. The second option is less efficient than the first one especially when that CFI function is called many times as a sequence of time consuming memory allocations and releasing would take place.
- The model_id is used as an input parameter in the EOCFI functions if it is needed.
- Close the model_id with **xl_model_close** (Only if the model_id was initialised).

Please refer also to:

- [MCD] for a detailed description of the models implemented for the Earth Observation CFI. (For the current version, only the default models are available)
- [GEN_SUM] for a detailed description of the *Id* concept.

5 LIBRARY INSTALLATION

For a detailed description of the installation of any CFI library, please refer to [GEN_SUM].

6 LIBRARY USAGE

The EO_LIB software library has the following dependencies:

- Other EOCFI libraries:
  - EO_FILE_HANDLING (See [F_H_SUM]).
  - EO_DATA_HANDLING (See [D_H_SUM]).
- Third party libraries:
  - POSIX thread library: libpthread.so (Note: this library is normally pre-installed in Linux and MacOS platforms. For Windows platforms, pthread.lib is included in the distribution package, with license LGPL);
  - GEOTIFF, TIFF, PROJ, LIBXML2 libraries (these libraries are included in the distribution package. Their usage terms and conditions are available in the file "TERMS_AND_CONDITIONS.TXT" which is part of the distribution package).

The following is required to compile and link a Software application that uses the EO_LIB software library functions (it is assumed that the required EOCFI and third-part libraries are located in directory *cfi_lib_dir* and the required header files are located in *cfi_include*, see [GEN_SUM] for installation procedures):

1) include the following header files in the source code:

   - explorer_lib.h (for a C application)

2) use the following compile and link options:

   Linux and MacOS platforms:

   -I*cfi_include_dir* -L*cfi_lib_dir* -lexplorer_lib

   - lexplorer_data_dandling -lexplorer_file_handling -lgeotiff -ltiff -lproj -lxml2 -lm -lc -lpthread

   Windows platforms:

   /I *"cfi_include_dir"* /libpath:*"cfi_lib_dir"* libexplorer_lib.lib

   libexplorer_data_handling.lib libexplorer_file_handling.lib libgeotiff.lib libtiff.lib libproj.lib libxml2.lib pthread.lib Ws2_32.lib

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

To avoid problems in linking a user application with the `EO_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 `XL_` or `xl_`.

It is possible to call the following CFI functions from a user application.

Table 1: CFI functions included within EO_LIB library

Function Name	Enumeration value	Long
Main CFI Functions		
xl_time_transport_to_ascii	XL_TIME_TRANSPORT_TO_ASCII_ID	0
xl_time_transport_to_transport	XL_TIME_TRANSPORT_TO_TRANSPORT_ID	1
xl_time_transport_to_processing	XL_TIME_TRANSPORT_TO_PROCESSING_ID	2
xl_time_processing_to_ascii	XL_TIME_PROCESSING_TO_ASCII_ID	3
xl_time_processing_to_transport	XL_TIME_PROCESSING_TO_TRANSPORT_ID	4
xl_time_processing_to_processing	XL_TIME_PROCESSING_TO_PROCESSING_ID	5
xl_time_ascii_to_ascii	XL_TIME_ASCII_TO_ASCII_ID	6
xl_time_ascii_to_transport	XL_TIME_ASCII_TO_TRANSPORT_ID	7
xl_time_ascii_to_processing	XL_TIME_ASCII_TO_PROCESSING_ID	8
xl_time_add	XL_TIME_ADD_ID	9
xl_time_diff	XL_TIME_DIFF_ID	10
xl_time_obt_to_time	XL_TIME_OBT_TO_TIME_ID	11
xl_time_time_to_obt	XL_TIME_TIME_TO_OBT_ID	12
xl_time_ref_init_file	XL_TIME_REF_INIT_FILE_ID	13
xl_time_ref_init	XL_TIME_REF_INIT_ID	14
xl_time_id_init	XL_TIME_ID_INIT	15
xl_time_ref_close	XL_TIME_CLOSE_ID	16
xl_change_cart_cs	XL_CHANGE_CART_CS_ID	17
xl_geod_to_cart	XL_GEOD_TO_CART_ID	18
xl_cart_to_geod	XL_CART_TO_GEOD_ID	19
xl_kepl_to_cart	XL_KEPL_TO_CART_ID	20
xl_cart_to_kepl	XL_CART_TO_KEPL_ID	21
xl_sun	XL_SUN_ID	22
xl_moon	XL_MOON_ID	23
xl_planet	XL_PLANET_ID	24
xl_star_radec	XL_STAR_RADEC_ID	25
xl_geod_distance	XL_GEOD_DISTANCE_ID	26
xl_time_get_leap_second_info	XL_TIME_GET_LEAP_SECOND_INFO_ID	27
xl_default_sat_init	XL_DEFAULT_SAT_INIT_ID	28

xl_run_init	XL_RUN_INIT_ID	29
xl_get_rotation_angles	XL_GET_ROTATION_ANGLES_ID	30
xl_get_rotated_vectors	XL_GET_ROTATED_VECTORS_ID	31
xl_position_on_orbit	XL_POSITION_ON_ORBIT	32
xl_quaternions_to_vectors	XL_QUATERNIONS_TO_VEC_ID	33
xl_vectors_to_quaternions	XL_VEC_TO_QUATERNIONS_ID	34
xl_star_catalog	XL_STAR_CATALOG_ID	35
xl_cart_to_radec	XL_CART_TO_RADEC_ID	36
xl_radec_to_cart	XL_RADEC_TO_CART_ID	37
xl_topocentric_to_ef	XL_TOPOCENTRIC_TO_EF_ID	38
xl_ef_to_topocentric	XL_EF_TO_TOPOCENTRIC_ID	39
xl_euler_to_matrix	XL_EULER_TO_MATRIX_ID	40
xl_matrix_to_euler	XL_MATRIX_TO_EULER_ID	41
xl_model_init_id	XL_MODEL_INIT_ID	42
xl_model_close	XL_MODEL_CLOSE_ID	43
xl_geoid_calc	XL_GEOID_CALC_ID	44
xl_quaternions_interpol	XL_QUATERNIONS_INTERPOL_ID	45
xl_time_cuc_to_processing	XL_TIME_CUC_TO_PROCESSING_ID	46
xl_time_processing_to_cuc	XL_TIME_PROCESSING_TO_CUC_ID	47
Error Handling Functions		
xl_verbose	not applicable	
xl_silent		
xl_get_code		
xl_get_msg		
xl_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 a 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

Every CFI function has a different length of the Error Vector, used in the calling I/F examples of this SUM and defined at the beginning of the library header file. In order to provide the user with a single value that could be used as Error Vector length for every function, a generic value has been defined (XL_ERR_VECTOR_MAX_LENGTH) as the maximum of all the Error Vector lengths. This value can therefore be safely used for every call of functions of this library.

6.2 General Enumerations

The aim of the current section is to present the enumeration values that can be used rather than integer parameters for some of the input parameters of the EO_LIB routines, as shown in the table below. The enumerations presented in [GEN_SUM] are also applicable.

Table 2: Enumerations within EO_LIB library

Input	Description	Enumeration value	Long
Time reference	Undefined	XL_TIME_UNDEF	-1
	TAI	XL_TIME_TAI	0
	UTC	XL_TIME_UTC	1
	UT1	XL_TIME_UT1	2
	GPS	XL_TIME_GPS	3
Processing format	Standard	XL_PROC	0
Transport time format	Standard	XL_TRANS_STD	0
	Envisat Ground Segment	XL_TRANS_ENVI_GS	11
	CryoSat He by the Segment	XL_TRANS_CRYO_GS	21
	CryoSat General Telemetry	XL_TRANS_CRYO_TM	22
	CryoSat SIRAL Telemetry	XL_TRANS_CRYO_TM_SIRAL	23
	SMOS transport time format	XL_TRANS_SMOS_TM	31
	GPS Second transport time format (number of seconds and microseconds elapsed since GPS epoch: 6th January 1980)	XL_TRANS_GENERIC_GPS_SEC	41
	GPS Week transport time format (number of weeks, seconds and microseconds elapsed since GPS epoch: 6th January 1980)	XL_TRANS_GENERIC_GPS_WEEK	42
ASCII time format	Undefined	XL_ASCII_UNDEF	-1
	Standard	XL_ASCII_STD	11
	Standard with reference	XL_ASCII_STD_REF	12
	Standard with microsecs	XL_ASCII_STD_MICROSEC	13
	Standard with reference and microsecs	XL_ASCII_STD_REF_MICROSEC	14
	Compact	XL_ASCII_COMPACT	21

Input	Description	Enumeration value	Long
	Compact with reference	XL_ASCII_COMPACT_REF	22
	Compact with microsecs	XL_ASCII_COMPACT_MICROSEC	23
	Compact with reference and microsecs	XL_ASCII_COMPACT_REF_MICROSEC	24
	Envisat	XL_ASCII_ENVI	31
	Envisat with reference	XL_ASCII_ENVI_REF	32
	Envisat with microsecs	XL_ASCII_ENVI_MICROSEC	33
	Envisat with reference and microsecs	XL_ASCII_ENVI_REF_MICROSEC	34
	CCSDS-A	XL_ASCII_CCSDSA	41
	CCSDS-A with reference	XL_ASCII_CCSDSA_REF	42
	CCSDS-A with microsecs	XL_ASCII_CCSDSA_MICROSEC	43
	CCSDS-A with reference and microsecs	XL_ASCII_CCSDSA_REF_MICROSEC	44
	CCSDS-A compact	XL_ASCII_CCSDSA_COMPACT	51
	CCSDS-A compact with reference	XL_ASCII_CCSDSA_COMPACT_REF	52
	CCSDS-A compact with microsecs	XL_ASCII_CCSDSA_COMPACT_MICROSEC	53
	CCSDS-A compact with reference and microsecs	XL_ASCII_CCSDSA_COMPACT_REF_MICROSEC	54
Time Initialization Mode	Initialization from file (data-driven)	XL_SEL_FILE	0
	Initialization within a time range	XL_SEL_TIME	1
	Initialization within a range of orbits	XL_SEL_ORBIT	2
	(not used in LIB)	XL_SEL_DEFAULT	3
Time Initialization Model	Select the file type automatically	XL_TIMEMOD_AUTO	-2
	User defined	XL_TIMEMOD_USER	-1
	None	XL_TIMEMOD_NONE	0
	IERS Bulletin B - Table 1 (Predicted)	XL_TIMEMOD_IERS_B_PREDICTED	1
	IERS Bulletin B - Table 2 (Restituted)	XL_TIMEMOD_IERS_B_RESTITUTED	2
	FOS Predicted Orbit File	XL_TIMEMOD_FOS_PREDICTED	3
	FOS Restituted Orbit File	XL_TIMEMOD_FOS_RESTITUTED	4
	DORIS Preliminary Orbit	XL_TIMEMOD_DORIS_PRELIMINARY	5
	DORIS Precise Orbit	XL_TIMEMOD_DORIS_PRECISE	6
	DORIS Navigator	XL_TIMEMOD_DORIS_NAVIGATOR	7
	Orbit Scenario File	XL_TIMEMOD_OSF	8
	IERS Bulletin A – Prediction table	XL_TIMEMOD_IERS_A_ONLY_PREDICTION	9
	IERS Bulletin A – Prediction table and extrapolation formula	XL_TIMEMOD_IERS_A_PREDICTION_AND_FORMULA	10
	IERS Bulletin B plus IERS Bulletin A (only prediction table for Bulletin A)	XL_TIMEMOD_IERS_B_AND_A_ONLY_PREDICTION	11
Reference frame	Barycentric Mean of 2000	XL_BM2000	1
	Heliocentric Mean of 2000	XL_HM2000	2

Input	Description	Enumeration value	Long
	Geocentric Mean of 2000	XL_GM2000	3
	Mean of Date	XL_MOD	4
	True of Date	XL_TOD	5
	Earth Fixed	XL_EF	6
	Launch Inertial Frame	XL_LIF	7
	Barycentric Mean of 1950	XL_BM1950	8
	Galactic Coordinates	XL_GALACTIC	9
Extended reference frames	Barycentric Mean of 2000.0	BAR_MEAN_2000	1
	Heliocentric Mean of 2000.0	HEL_MEAN_2000	2
	Geocentric Mean of 2000.0	GEO_MEAN_2000	3
	Mean of date	MEAN_DATE	4
	True of date	TRUE_DATE	5
	Pseudo Earth Fixed	PSEUDO_EARTH_FIXED	6
	Earth Fixed	EARTH_FIXED	7
	Launch Inertial Frame	LIF	8
	Barycentric Mean of 1950	BAR_MEAN_1950	9
	Galactic Coordinates	GALACTIC	10
	Satellite relative actual reference cs	SAT_ACT_REF	11
	Quasi-Mean of Date	QUASI_MEAN_DATE	12
	Pseudo-True of Date	PSE_TRUE_DATE	13
	Topocentric coordinate system	TOPOCENTRIC	14
	Satellite reference frame	SAT_REF	15
	Satellite relative reference frame	SAT_REL_REF	16
Kepler OSV mode	Mean Kepler State Vector	XL_KEPLER_MEAN	1
	Osculating Kepler State Vector	XL_KEPLER_OSC	2
Planet ID	Mercury	XL_MERCURY	1
	Venus	XL_VENUS	2
	Earth-Moon barycenter	XL_EM_BAR	3
	Mars	XL_MARS	4
	Jupiter	XL_JUPITER	5
	Saturn	XL_SATURN	6
	Uranus	XL_URANUS	7
	Neptune	XL_NEPTUNE	8
Calculation mode	Position (using Bowring iterative method for xl_cart_to_geod)	XL_CALC_POS	1
	Position and velocity (using Bowring iterative method for xl_cart_to_geod)	XL_CALC_POS_VEL	2
	Position, velocity and acceleration	XL_CALC_POS_VEL_ACC	3
	Position (using Bowring iterative method for xl_cart_to_geod)	XL_CALC_ITER_POS	4
	Position and velocity (using Bowring	XL_CALC_ITER_POS_VEL	5

Input	Description	Enumeration value	Long
	iterative method for xl_cart_to_geod)		
	Position (using Bowring non iterative method for xl_cart_to_geod)	XL_CALC_NO_ITER_POS	6
	Position and velocity (using Bowring non iterative method for xl_cart_to_geod)	XL_CALC_NO_ITER_POS_VEL	7
AOCS mode	Default Cx, Cy, Cz values	XL_AOCS_DEFAULT	0
	User defined Cx, Cy, Cz values	XL_AOCS_USER	1
	Geocentric pointing	XL_AOCS_GPM	2
	Local normal pointing	XL_AOCS_LNP	3
	Yaw steering + local normal pointing	XL_AOCS_YSM	4
Angle Type	True Latitude (TOD)	XL_ANGLE_TYPE_TRUE_LAT_TOD	1
	Mean Latitude (TOD)	XL_ANGLE_TYPE_MEAN_LAT_TOD	2
	True Latitude (GM2000)	XL_ANGLE_TYPE_TRUE_LAT_GM2000	3
	True Latitude (EF)	XL_ANGLE_TYPE_TRUE_LAT_EF	4
Derivatives	No derivative	XL_NO_DER	0
	First in his joy is also calculated	XL_DER_1ST	1
	First and second derivative.	XL_DER_2ND	2
Type of *Ids*	Unknown	XL_INIT_UNKNOWN	0
	runId	XL_INIT_RUN	1
	timeId	XL_INIT_TIME	2
	orbitId (not used in LIB)	XO_INIT_ORBIT	3
	propagId (not used in LIB)	XO_INIT_PROPAG	4
	interpolId (not used in LIB)	XO_INIT_INTERPOL	5
	sat_nom_att_Id (not used in LIB)	XP_INIT_SAT_NOM_ATT	6
	sat_att_Id (not used in LIB)	XP_INIT_SAT_ATT	7
	instr_att_Id (not used in LIB)	XP_INIT_INSTR_ATT	8
	attitudeId (not used in LIB)	XP_INIT_ATTITUDE	9
	atmosId (not used in LIB)	XP_INIT_ATMOS	10
	demId (not used in LIB)	XP_INIT_DEM	11
	targetId (not used in LIB)	XP_INIT_TARGET	12
Boolean values	False	XL_FALSE	0
	True	XL_TRUE	1
Star Catalogues	FK4 Star catalogue	XL_FK4	0
	FK5 Star catalogue	XL_FK5	1
Vector mode flag	Point location	XL_MODE_FLAG_LOCATION	0
	Direction vector	XL_MODE_FLAG_DIRECTION	1
Model sets	CFI Default models	XL_MODEL_DEFAULT	0
	User defined models	XL_MODEL_CONFIG	1
Model types	Earth model	XL_MODEL_TYPE_EARTH	0

Input	Description	Enumeration value	Long
	Sun model	XL_MODEL_TYPE_SUN	1
	Moon model	XL_MODEL_TYPE_MOON	2
	Planet model	XL_MODEL_TYPE_PLANET	3
	Star model	XL_MODEL_TYPE_STAR	4
	Nutation model	XL_MODEL_TYPE_NUTATION	5
	Precession model	XL_MODEL_TYPE_PRECESSION	6
	Constant model	XL_MODEL_TYPE_CONSTANTS	7
	Light propagation model	XL_MODEL_TYPE_LIGHT_PROPAGATION	8
	Number of models	XL_NUM_MODEL_TYPES_ENUM	9
Earth model	Earth Default model	XL_MODEL_EARTH_DEFAULT	0
Sun model	Sun Default model	XL_MODEL_SUN_DEFAULT	0
		XL_MODEL_SUN_TRAVEL_TIME	1
Moon model	Moon Default model	XL_MODEL_MOON_DEFAULT	0
Planet model	Planet Default model	XL_MODEL_PLANET_DEFAULT	0
Star model	Star Default model	XL_MODEL_STAR_DEFAULT	0
Nutation model	Nutation Default model	XL_MODEL_NUTATION_DEFAULT	0
Precession model	Precession Default model	XL_MODEL_PRECESSION_DEFAULT	0
Constants model	Contants Default model	XL_MODEL_CONSTANTS_DEFAULT	0
Light propagation model	Default light propagation mode. Light travel time is not taken into account.	XL_MODEL_LIGHT_PROPAGATION_DISABLED	0
	The target functions keep into account the time spent by a generic signal travelling at the speed of light to go from the target to the satellite	XL_MODEL_LIGHT_PROPAGATION_RECEIVER	1
	The target functions keep into account the time spent by a generic signal travelling at the speed of light go from the satellite to the target	XL_MODEL_LIGHT_PROPAGATION_TRANSMITTER	2
Bulletin type	File is not an IERS Bulletin	XL_NO_BULLETIN	0
	IERS Bulletin B	XL_BULLETIN_B	1
	IERS Bulletin A	XL_BULLETIN_A	2
	IERS Bulletin B plus IERS Bulletin A	XL_BULLETIN_B_AND_A	3
Extrapolation formulas activation	IERS A Extrapolation formulas enabled	XL_FORMULA_ENABLED	0
	IERS A Extrapolation formulas disabled	XL_FORMULA_DISABLED	1
Data origin for time initialization with xl_time_id_init	Data read from file	XL_FILE_DATA	0
	Data from user time correlation data	XL_TIME_CORRELATIONS_DATA	1
List of algorithms that	Slerp interpolation (see details:	XL_INTERPOL_SLERP	0

Input	Description	Enumeration value	Long
can be used for quaternions interpolation	http://en.wikipedia.org/wiki/Slerp)		
CUC time type	Only T field	XL_CUC_T_FIELD	0
	P field and T field	XL_CUC_T_AND_P_FIELDS	1
CUC epoch type	Use CCSDS epoch, 01/01/1958, 00h00	XL_EPOCH_CCSDS	0
	Use GPS epoch, 6 of January 1986, 00h00	XL_EPOCH_GPS	1
	Epoch taken from user input	XL_EPOCH_USER_DEFINED	2

The use of the previous enumeration values could be restricted by the particular usage within the different CFI functions. The actual range to be used is indicated within a dedicated reference named ***allowed range***. When there are not restrictions to be mentioned, the allowed range column is populated with the label ***complete***.

The meanings and units of the different array elements from the Transport time strongly depend upon the selected Transport format (by means of the Transport format ID). The table below shows the choices:

Table 3: Transport time formats

Input	Array Element	Unit and shun)	Allowed Range
XL_TRANS_STD	[0]	Integer days	[-18262,36524]
	[1]	Integer seconds	[0,86399]
	[2]	Integer microseconds	[0,999999]
XL_TRANS_ENVI_GS	[0]	Integer days	[-18262,36524]
	[1]	Integer seconds	[0,86399]
	[2]	Integer microseconds	[0,999999]
XL_TRANS_CRYO_GS	[0]	Integer days	[-18262,36524]
	[1]	Integer seconds	[0,86399]
	[2]	Integer microseconds	[0,999999]
XL_TRANS_CRYO_TM	[0]	Integer days	[-18262,36524]
	[1]	Integer milliseconds	[0,86399999]
	[2]	Integer microseconds	[0,999]
XL_TRANS_CRYO_TM_SIRAL	[0]	Integer days	[-18262,36524]
	[1]	Integer milliseconds	[0,86399999]
	[2]	Integer microseconds	[0,999]
	[3]	SIRAL extra counter	[0,1745454545]

Input	Array Element	Unit and shun)	Allowed Range
XL_TRANS_SMOS_TM	[0]	Week number	[-1566, 6260]
	[1]	Seconds of week	[0, 604799]
	[2]	Fraction of seconds	[0, 65535]
XL_TRANS_GENERIC_GPS_SEC	[0]	Number of seconds	[0, INT_MAX] Note: INT_MAX is a macro that returns the maximum number a C int variable can store
	[1]	Number of microseconds	[0, 999999]
XL_TRANS_GENERIC_GPS_WEEK	[0]	Number of weeks	[0, 6260]
	[1]	Number of seconds	[0, 604799]
	[2]	Number of microseconds	[0, 999999]

The string characteristics of the ASCII time formats depends strongly upon the selected ASCII format (by means of the ASCII format ID). The tables below show the available choices:

Note that the value of 86400 for seconds (and 86400000 for milliseconds) is accepted only for UTC in case a leap second is being introduced. This may happen only at 23:59 minutes and only on four days of the year (31/03, 30/06, 30/09, 31/12). The decision to introduce a leap second in UTC is the responsibility of the International Earth Rotation Service (IERS). See [IERS] for further details.

For further details on the SIRAL extra counter for the Cryosat mission please see [MCD].

Table 4: Basic ASCII time formats

Input	String format
XL_ASCII_UNDEF	-
XL_ASCII_STD	“yyyy-mm-dd_hh:nn:ss“
XL_ASCII_COMPACT	“yyyymmdd_hhnnss“
XL_ASCII_ENVI	“dd-mmm-yyyy hh:nn:ss”
XL_ASCII_CCSDSA	“yyyy-mm-ddThh:nn:ss“
XL_ASCII_CCSDSA_COMPACT	“yyyymmddThhnnss“

Table 5: Derived ASCII time formats

Input	String format
XL_ASCII_STD_REF	“RRR=yyyy-mm-dd_hh:nn:ss“
XL_ASCII_STD_MICROSEC	“yyyy-mm-dd_hh:nn:ss.uuuuuu“
XL_ASCII_STD_REF_MICROSEC	“RRR=yyyy-mm-dd_hh:nn:ss.uuuuuu“
XL_ASCII_COMPACT_REF	“RRR=yyyymmdd_hhnnss“
XL_ASCII_COMPACT_MICROSEC	“yyyymmdd_hhnnssuuuuuu“
XL_ASCII_COMPACT_REF_MICROSEC	“RRR=yyyymmdd_hhnnssuuuuuu“
XL_ASCII_ENVI_REF	“RRR=dd-mmm-yyyy hh:nn:ss”

Input	String format
XL_ASCII_ENVI_MICROSEC	“dd-mmm-yyyy hh:nn:ss.uuuuuu”
XL_ASCII_ENVI_REF_MICROSEC	“RRR=dd-mmm-yyyy hh:nn:ss.uuuuuu”
XL_ASCII_CCSDSA_REF	“RRR=yyyy-mm-ddThh:nn:ss“
XL_ASCII_CCSDSA_MICROSEC	“yyyy-mm-ddThh:nn:ss.uuuuuu“
XL_ASCII_CCSDSA_REF_MICROSEC	“RRR=yyyy-mm-ddThh:nn:ss.uuuuuu“
XL_ASCII_CCSDSA_COMPACT_REF	“RRR=yyyymmddThhnnss“
XL_ASCII_CCSDSA_COMPACT_MICROSEC	“yyyymmddThhnnssuuuuuu“
XL_ASCII_CCSDSA_COMPACT_REF_MICROSEC	“RRR=yyyymmddThhnnssuuuuuu“

where:

- *yyyy* stands for the year
- *mm* stands for the month expressed as a numerical count, i.e. 01 for January, etc
- *mmm* stands for the month expressed in abbreviatures, i.e. JAN, MAR, etc
- *dd* stands for the day of month
- *ddd* stands for the day of the year
- *hh* stands for the hour in the day
- *nn* stands for the minutes within a hour
- *ss* stands for the seconds within a minute
- *uuuuuu* stands for the microseconds within a second
- *RRR* stands for the time reference (TAI, UTC, UT1 or GPS)

In ASCII formats two values are defined, by convention, as Beginning of Mission (BOM) and End of Mission (EOM). These values are listed, for the various ASCII time formats, inTable 6 and Table 7.

Usually a date with all zeros is seen as EOM, and a date with all nines is considered EOM. The only exception are the ENVISAT-specific formats, which use as EOM the date December 31st, 2078 at 23:59:59.999999.

Format transformations of BOM and EOM between ASCII format is allowed.

Time reference is not considered in BOM or EOM, thus any time reference is accepted (TAI, UTC, UT1 or GPS) for the values in Table 6 and Table 7..

BOM and EOM do not have an equivalent in Processing or Transport formats, so if the user tries to convert them from ASCII to another non-ASCII format an error will occur.

Table 6: Definition of BOM and EOM for basic ASCII time formats

ASCII format	Beginning of Mission	End of Mission
XL_ASCII_UNDEF	-	-
XL_ASCII_STD	“0000-00-00_00:00:00“	“9999-99-99_99:99:99“
XL_ASCII_COMPACT	“00000000_000000“	“99999999_999999“
XL_ASCII_ENVI	“00-000-0000_00:00:00”	“31-DEC-2078 23:59:59”
XL_ASCII_CCSDSA	“0000-00-00T00:00:00“	“9999-99-99T99:99:99“

ASCII format	Beginning of Mission	End of Mission
XL_ASCII_CCSDSA_COMPACT	“00000000T000000“	“99999999T999999“

Table 7: Definition of BOM and EOM for derived ASCII time formats

ASCII format	Beginning of Mission	End of Mission
XL_ASCII_STD_REF	“RRR=0000-00-00_00:00:00“	“RRR=9999-99-99_99:99:99“
XL_ASCII_STD_MICROSEC	“0000-00-00_00:00:00.000000“	“9999-99-99_99:99:99.999999“
XL_ASCII_STD_REF_MICROSEC	“RRR=0000-00-00_00:00:00.000000“	“RRR=9999-99-99_99:99:99.999999“
XL_ASCII_COMPACT_REF	“RRR=00000000_000000“	“RRR=99999999_999999“
XL_ASCII_COMPACT_MICROSEC	“00000000_000000000000“	“99999999_999999999999“
XL_ASCII_COMPACT_REF_MICROSEC	“RRR=00000000_000000000000“	“RRR=99999999_999999999999“
XL_ASCII_ENVI_REF	“RRR=00-000-0000_00:00:00”	“RRR=31-DEC-2078 23:59:59”
XL_ASCII_ENVI_MICROSEC	“00-000-0000_00:00:00.000000”	“31-DEC-2078 23:59:59.999999”
XL_ASCII_ENVI_REF_MICROSEC	“RRR=00-000-0000_00:00:00.000000”	“RRR=31-DEC-2078 23:59:59.999999”
XL_ASCII_CCSDSA_REF	“RRR=0000-00-00T00:00:00“	“RRR=9999-99-99T99:99:99“
XL_ASCII_CCSDSA_MICROSEC	“0000-00-00T00:00:00.000000“	“9999-99-99T99:99:99.999999“
XL_ASCII_CCSDSA_REF_MICROSEC	“RRR=0000-00-00T00:00:00.000000“	“RRR=9999-99-99T99:99:99.999999“
XL_ASCII_CCSDSA_COMPACT_REF	“RRR=00000000T000000“	“RRR=99999999T999999“
XL_ASCII_CCSDSA_COMPACT_MICROSEC	“00000000T000000000000“	“99999999T999999999999“

where:

- *RRR* stands for the time reference (TAI, UTC, UT1 or GPS)

6.3 Data Structures

The aim of the current section is to present the data structures that are used in the EO_LIB library. The structures are currently used for the CFI Identifiers accessor functions. The following table show the structures with their names and the data that contain:

Table 8: EO_LIB structures

Structure name	Data		
	Variable Name	C type	Description
xl_par_der	deriv	XL_Deriv_enum	Flag to indicate if the 1st and 2nd derivatives are defined
	p	double	The parameter, expressed in the appropiate units
	pd	double	1st time derivative of the parameter
	p2d	double	2nd time derivative of the parameter
xl_cord	cs	XL_CS_rl_enum	Coordinate reference frame
	deriv	XL_Deriv_enum	Flag to indicate if the 1st and 2nd derivatives are defined
	v	double [3]	Vector
	vd	double [3]	Vector rate
	v2d	double [3]	Vector rate-rate
xl_cuc_time_config	cuc_type	long	CUC time type (see enumeration CUC time type in section 6.2)
	cuc_epoch	long	CUC epoch type (see enumeration CUC epoch type in section 6.2)
	time_ref	long	Time reference of the epoch (see time reference enumeration in section 6.2)
	epoch	double	Reference epoch provided by user.
	basic_time_unit_num_octets	long	Number of basic time unit octets of CUC time
	fractional_time_unit_num_octets	long	Number of fraction time units octets of CUC time
xl_cs_tra	azel_flag	XL_Boolean	Flag to indicate if an azimuth/elevation definition has been provided.
	azel_def	xl_az_el_definition	Azimuth/elevation definition
	ref_i	XL_CS_rl_enum	Initial reference frame
	ref_f	XL_Attitude_fr_enum	final reference frame
	amb_flag	XL_Boolean	Ambiguity flag
	deriv	XL_Deriv_enum	Flag to indicate if the 1st and 2nd derivatives are defined
	v	double [3]	Translation vector from ref_i to ref_f
	vd	double [3]	Translation rate vector from ref_i to ref_f
	v2d	double [3]	Translation rate-rate vector from ref_i to ref_f
	m	double [3][3]	Rotation matrix from ref_i to ref_f
	md	double [3][3]	Rotation matrix rate from ref_i to ref_f
	m2d	double [3][3]	Rotation matrix rate-rate from ref_i to ref_f

Structure name	Data		
	Variable Name	C type	Description
xl_par_der	deriv	XL_Deriv_enum	Flag to indicate if the 1st and 2nd derivatives are defined
	p	double	The parameter, expressed in the appropiate units
	pd	double	1st time derivative of the parameter
	p2d	double	2nd time derivative of the parameter
xl_time_correlations	tai_time	double	TAI time
	ut1_time	double	UT1 time
	tai_utc	double	difference between TAI and UTC
	tai_ut1	double	difference between TAI and UT1
	tai_gps	double	difference between TAI and GPS
xl_leap_second	flag	long	XL_TRUE if the leap second exists
	utc_time	double	UTC time for the leap second
xl_time_id_data	iers_bulletin_type	long	Bulletin type (see Bulletin type)
	iers_formula_flag	long	IERS extrapolation formula enabled or disabled (see Extrapolation formulas activation)
	prediction_first_record	long	Indicates the 1st record belonging to bulletin A if applicable (starting at 0).
	polar_motion_formula	xl_polar_motion_formula	Polar motion formula parameters
	time_correlation_formula	xl_time_correlation_formula	Time correlation formula parameters
	num_lines	long	Number of records in the array with the time correlations
	time_str	xl_time_correlations*	Array with the time correlations
	polar_motion_params	xl_polar_motion_params*	Array with the polar motion parameters (it can be NULL if there are no parameters in initialization)
	leap_sec	xl_leap_second	Leapsecond information
	launch_inertial_frame_config	xl_launch_inertial_frame_config	Launch inertial frame configuration
xl_az_el_definition	az_0_axis	long	Azimuth 0deg axis (one of the values in XL_Axis_enum)
	az_90_axis	long	Azimuth 90deg axis (one of the values in XL_Axis_enum)
	el_90_axis	long	Elevation 90deg axis (one of the values in XL_Axis_enum)

Structure name	Data		
	Variable Name	C type	Description
xl_par_der	deriv	XL_Deriv_enum	Flag to indicate if the 1st and 2nd derivatives are defined
	p	double	The parameter, expressed in the appropiate units
	pd	double	1st time derivative of the parameter
	p2d	double	2nd time derivative of the parameter
xl_model_data	earth_model	long	Earth model
	sun_model	long	Sun model
	moon_model	long	Moon model
	planet_model	long	Planets model
	star_model	long	Stars model
	nutation_model	long	Nutation model
	precession_model	long	Precession model
	constants_model	long	Constants model
	re	double	Earth equatorial radius [m]
	mu	double	Earth's gravitational constant [m3/s2]
	j2	double	Second zonal harmonic
	j3	double	Third zonal harmonic
	j4	double	Fourth zonal harmonic
	major_axis	double	Semi-major axis [m]
	minor_axis	double	Semi-minor axis [m]
	ecc	double	First eccentricity [-]
	flat	double	Flattening [-]
	gcoef_0	double	Greenwich sidereal angle for t=0 (MJD 2000)
	gcoef_1	double	1st. Derivative of the Greenwich sidereal angle for t=0 (MJD 2000)
	gcoef_2	double	2nd. Derivative of the Greenwich sidereal angle for t=0 (MJD 2000)
	gcoef_sim_0	double	Greenwich sidereal angle for t=0 (MJD 2000) (Simplified model)
	gcoef_sim_1	double	1st. Derivative of the Greenwich sidereal angle for t=0 (MJD 2000) (Simplified model)
	gcoef_sim_2	double	2nd. Derivative of the Greenwich sidereal angle for t=0 (MJD 2000) (Simplified model)
	au	double	Astronomical units in kms
xl_polar_motion_params	x	double	x-axis is in the direction of the IERS Reference Meridian (IRM)
	y	double	y-axis is in the direction 90 degrees West longitude

Structure name	Data		
	Variable Name	C type	Description
xl_par_der	deriv	XL_Deriv_enum	Flag to indicate if the 1st and 2nd derivatives are defined
	p	double	The parameter, expressed in the appropiate units
	pd	double	1st time derivative of the parameter
	p2d	double	2nd time derivative of the parameter
xl_polar_motion_formula	ax	double	x parameter formula: constant term
	bx	double	x parameter formula: cos(A) coefficient
	cx	double	x parameter formula: sin(A) coefficient
	dx	double	x parameter formula: cos(C) coefficient
	ex	double	x parameter formula: sin(C) coefficient
	ay	double	y parameter formula: constant term
	by	double	y parameter formula: cos(A) coefficient
	cy	double	y parameter formula: sin(A) coefficient
	dy	double	y parameter formula: cos(C) coefficient
	ey	double	y parameter formula: sin(C) coefficient
	A_ref	double	Reference day for A parameter formula (MJD2000)
	A_div	double	Divisor for A parameter formula
	C_ref	double	Reference day for C parameter formula (MJD2000)
	C_div	double	Divisor for C parameter formula
xl_time_correlation_formula	a	double	Constant parameter
	b	double	Linear coefficient
	b_ref	double	Reference day (MJD2000)
xl_time_id_init_data_union	file_set	xd_eocfi_file_set	Set of data from files
	time_id_data	xl_time_id_data	Data for time correlation initialization
xl_time_id_init_data	data_type	long	Enumeration value from xl_time_data_origin_enum: XL_FILE_DATA or XL_TIME_CORRELATION_DATA
	time_id_init_data	xl_time_id_init_data_union	Data for time initialization
xl_geoid_calc_inputs	model_id	xl_model_id*	Model to be used in computation.
	latitude	double	Latitude [-90., 90.] [deg]
	longitude	double	Longitude [0., 360.) [deg]
	utc_time	double	UTC time (processing format)
	nof_harmonics	long	Number of harmonics to be used.
xl_geoid_calc_outputs	undulation	double	Height of the geoid over the ellipsoid [m]
xl_quaternions_interpol_cfg	algo	long	Algorithm to be used for interpolation. See xl_quaternions_interpol_algo_enum in table 2 for possible values

Structure name	Data		
	Variable Name	C type	Description
xl_par_der	deriv	XL_Deriv_enum	Flag to indicate if the 1st and 2nd derivatives are defined
	p	double	The parameter, expressed in the appropiate units
	pd	double	1st time derivative of the parameter
	p2d	double	2nd time derivative of the parameter
xl_launch_inertial_frame_config	enabled_flag	long	This flag indicates if the conversion from/to LIF frame is enabled (XL_TRUE) or disabled (XL_FALSE).
	longitude	double	The longitude of the reference meridian for LIF reference frame (see [MCD])
	utc_time	double	The reference UTC time (see [MCD]).

7 CFI FUNCTIONS DESCRIPTION

The following sections describe each CFI function.

The calling interfaces are described for C 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])

7.1 xl_time_ref_init_file

7.1.1 Overview

The **xl_time_ref_init_file** CFI function initializes time correlations between TAI, UTC, UT1 and GPS times from reference data files. The correlations provided by the different input files can be found in the following table (for details about file formats, see [D_H_SUM]).

Table 9: Time reference correlations from reference files

	TAI	UTC	UT1	GPS	orbit
FOS Predicted Orbit File	X	X	X	(x)	X
FOS Restituted Orbit File	X	X	X	(x)	X
DORIS Preliminary Orbit	X	X	X	(x)	X
DORIS Precise Orbit	X	X	X	(x)	X
DORIS Navigator File	X	X	X	(x)	X
IERS Bulletin B format 1980	X	X	X	(x)	
IERS Bulletin B format 2010	X	X	X	(x)	
Orbit Scenario File	X	X	X	(x)	
IERS Bulletin A	X	X	X	(X)	
IERS Bulletin B plus IERS Bulletin A	X	X	X	(X)	

Normally a single Predicted or DORIS Orbit file is sufficient to have all correlations needed (the (x) mark indicates that the GPS time correlation, although is not present within the file, can be simulated since it is always a fixed delta from TAI). The last updated IERS Bulletins can be downloaded from IERS bulletins web page ([IERS]).

When using an Orbit Scenario File, it must be taken into account that, since one orbital change can be far away from the following one, leap seconds could be calculated wrongly if there is more than one of the four possible leap second insertion points (end of March, end of June, end of September and end of December) between them.

When using a Bulletin B and a Bulletin A for initialization (B plus A initialization) it must be taken into account that:

•The first file in the input list must be Bulletin B, not Bulletin A.

•For Bulletin B, FINAL and PRELIMINARY tables are used; for Bulletin A, PREDICTION table is used.

•First record of Bulletin B must be before first record of Bulletin A.

•Last record of Bulletin A must be after last record of Bulletin B.

•In case of partial overlap (last record of Bulletin B is after first record of Bulletin A), the records of Bulletin A loaded are the ones that are after the last record of Bulletin B).

•In case of no overlap (first record of Bulletin B is after last record of Bulletin A), the gap must be less than 1 month (which is the periodicity of Bulletin B).

•IERS bulletins contains polar motion parameters. These parameters are stored in the time_id and are used for co-ordinate system conversions. When the bulletin is not used, polar motion parameters are set to zero.

All other input files are ESA-provided. These initialization files could even be generated by the users by means of EO_FILE_HANDLING and EO_DATA_HANDLING CFI libraries.

In case multiple files are used for the time correlations initializations, the files should be time ordered. If there is overlap between files, the newest data have precedence.

For Orbit Scenario File, only one file is admitted. If more files are introduced a warning is raised and the computations are performed only with the first OSF introduced.

A complete calling sequence of the time reference computations is presented in section 4.2.

The validity interval of the initialization depends on the input file, according to the following table:

Table 10: Initialization validity depending on input file

	Validity start	Validity stop
FOS Predicted Orbit File FOS Restituted Orbit File DORIS Preliminary Orbit DORIS Precise Orbit DORIS Navigator File	Time of the first state vector in input files that belongs to input range	Time of the last state vector in input files that belongs to input range
ERS Bulletin B format 1980 IERS Bulletin B format 2010	Time of first record in tables FINAL or PRELIMINARY (for PREDICTED initialization) or table SMOOTHED (for RESTITUTED initialization) that belongs to input range	Time of last record in tables FINAL or PRELIMINARY (for PREDICTED initialization) or table SMOOTHED (for RESTITUTED initialization) that belongs to input range
IERS Bulletin A (only prediction)	Time of first record in PREDICTION table that belongs to input range	Time of last record in PREDICTION table that belongs to input range
IERS Bulletin A (prediction and formula)	Time of first record in PREDICTION table that belongs to input range	End of Mission
IERS Bulletin B plus IERS Bulletin A	Time of first record in FINAL or PRELIMINARY table of Bulletin B that belongs to input range	Time of last record in PREDICTION table of Bulletin A that belongs to input range
Orbit Scenario File	Time of first orbital change in file	End of mission

If any time operation is done with a time outside validity interval, the behaviour of the functions is the following:

•If the time is lower than the start validity time, a warning is returned and the time correlation used for the time computations is that of the first record stored in xl_time_id in the initialization.

•If the time is greater than the stop validity time, a warning is returned and the time correlation used for the time computations is that of the last record stored in xl_time_id in the initialization, except for the case of IERS Bulletin A prediction plus formula initialization, where the extrapolation formulas are used.

In order to read files, xl_time_ref_init_file function internally uses Data Handling functions. Please refer to [D_H_SUM] in particular sections 4.2 and 4.3, for further details.

7.1.2 Calling interface

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

```
#include <explorer_lib.h>
{
      long time_model, n_files, time_init_mode, time_ref;
      long orbit0, orbit1;
      char **time_file;
      double time0, time1, val_time0, val_time1;
      xl_time_id time_id = {NULL};
      long ierr[XL_NUM_ERR_TIME_REF_INIT_FILE], status;

      status = xl_time_ref_init_file (&time_model, &n_files,
                                      time_file, &time_init_mode,
                                      &time_ref, &time0, &time1,
                                      &orbit0, &orbit1,
                                      &val_time0,&val_time1,
                                      &time_id, ierr);
}
```

7.1.3 Input parameters

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

Table 11: Input parameters of xl_time_ref_init_file function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
time_model	long *	-	Time model ID	-	Complete except XL_TIMEMOD_USER Note: When the input file is an IERS Bulletin B format 1980 file and the time mode is XL_TIMEMOD_AUTO, then the time model is set automatically to XL_TIMEMOD_IERS_B_RESTITUTED Note: When the input file is an IERS Bulletin B format 2010, the time models XL_TIMEMOD_IERS_B_RESTITUTED and XL_TIMEMOD_IERS_B_PRED

					ICTED coincide. Note: Whe the input file is an IERS Bulletin A and the time mode is XL_TIMEMOD_AUTO, the the time model is set automatically to XL_TIMEMOD_IERS_A_PREDICTION_AND_FORMULA
n_files	long *	-	Number of reference data files	-	> 0
time_file	char**	-	Filenames of the reference data files	-	-
time_init_mode	long *	-	Flag for selecting the time range of the initialisation: It could be the whole file (*XL_SEL_FILE*), the orbit range given by orbit0-orbit1 (*XO_SEL_ORBIT*) or the time range given by time0-time1(*XO_SEL_TIME*)	-	Select either: · XL_SEL_FILE · XL_SEL_ORBIT · XL_SEL_TIME - XL_SEL_ORBIT is not allowed for IERS Bulletins (any format) nor DORIS Navigator files - XL_SEL_ORBIT and XL_SEL_TIME are not enabled for OSF
time_ref	long *	-	Time reference ID	-	Complete. If the input file is a DORIS Navigator file and the time_init_mode is XL_SEL_TIME, then only time_ref allowed is XL_TIME_UTC.
time0	double*	-	If: *time_init_mode=XL_SEL_TIME* Start of the time range defined by [time0,time1]	Decimal days (Processing format)	[-18262.0,36524.0]
time1	double*	-	If: *time_init_mode=XL_SEL_TIME* End of the time range defined by [time0,time1]	Decimal days (Processing format)	[-18262.0,36524.0] > time0
orbit0	long*	-	If: *time_init_mode=XL_SEL_ORBIT* Absolute orbit number corresponding to the start of the time range defined by [ANX_{orbit0}, $ANX_{orbit1+1}$]	-	>= 0
orbit1	long*	-	If: *time_init_mode=XL_SEL_ORBIT* Absolute orbit number corresponding to the end of the time range defined by [ANX_{orbit0}, $ANX_{orbit1+1}$]	-	>orbit0

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

- Time model ID: time_model. See current document, section Error: Reference source not found "Time Initialization Model" enumeration.
- Time reference ID: time_ref. See current document, section Error: Reference source not found "Time reference" enumeration .
- Time range initialisation flag: time_init_mode. See current document, section 6.2.

7.1.4 Output parameters

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

Table 12: Output parameters of xl_time_ref_init_file function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_time_ref_init_file	long	-	Status flag	-	-
val_time0	double*	-	Validity start time of the initialization (see table 10).	Decimal days (Processing format)	[-18262.0,36524.0]
val_time1	double*	-	Validity end time of the initialization (see table 10).	Decimal days (Processing format)	[-18262.0,36524.0]
time_id	xl_time_id*	-	Structure that contains the time correlations.	-	-
ierr	long	-	Error vector	-	-

Note that ***val_time0*** and ***val_time1*** can define a validity range different to that requested by the user. This range gives the maximum coverage provided by the input files within the margins selected by the user (see table 10).

It has to be remarked that if the input time is outside the range of initialization, transformations are performed anyway, using the closest correlation data (or the extrapolation formula for IERS Bulletin A prediction plus formula initialization). However a warning is returned, since there is no guarantee that the correlation is correct.

7.1.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_time_ref_init_file** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 error vector returned by the **xl_time_ref_init_file** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM])

Table 13: Error messages of xl_time_ref_init_file function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Time model ID is not correct	No calculation performed	XL_CFI_TIME_REF_INIT_FILE_TIME_MODEL_ERR	0
ERR	Non-positive number of data files	No calculation performed	XL_CFI_TIME_REF_INIT_FILE_N_FILES_ERR	1
ERR	Incorrect file names	No calculation performed	XL_CFI_TIME_REF_INIT_FILE_FILE_NAMES_ERR	2
ERR	Time init mode ID is not correct	No calculation performed	XL_CFI_TIME_REF_INIT_FILE_INIT_MODE_ERR	3
ERR	Time reference ID is not correct	No calculation performed	XL_CFI_TIME_REF_INIT_FILE_TIME_ERR	4
ERR	Reference start time out of limits	No calculation performed	XL_CFI_TIME_REF_INIT_FILE_DAY_0_ERR	5
ERR	Reference end time out of limits	No calculation performed	XL_CFI_TIME_REF_INIT_FILE_DAY_1_ERR	6
ERR	Wrong reference time range	No calculation performed	XL_CFI_TIME_REF_INIT_FILE_DAY_RANGE_ERR	7
ERR	Reference start orbit is negative	No calculation performed	XL_CFI_TIME_REF_INIT_FILE_ORB_0_ERR	8
ERR	Reference end orbit is negative	No calculation performed	XL_CFI_TIME_REF_INIT_FILE_ORB_1_ERR	9
ERR	Wrong reference orbit range	No calculation performed	XL_CFI_TIME_REF_INIT_FILE_ORB_RANGE_ERR	10
ERR	File does not exist	No calculation performed	XL_CFI_TIME_REF_INIT_FILE_FILE_ERR	11
ERR	Time table is empty or has wrong format	No calculation performed	XL_CFI_TIME_REF_INIT_FILE_TABLE_ERR	12
ERR	Time range from file is outside input range	No calculation performed	XL_CFI_TIME_REF_INIT_FILE_TIME_OUTSIDE_RANGE_ERR	13
ERR	Orbit range from file is outside input range	No calculation performed	XL_CFI_TIME_REF_INIT_FILE_ORB_OUTSIDE_RANGE_ERR	14
ERR	Memory allocation error	No calculation performed	XL_CFI_TIME_REF_INIT_FILE_MEMORY_ERR	15
ERR	Error in reading file	No calculation performed	XL_CFI_TIME_REF_INIT_FILE_READ_FILE_ERR	16
ERR	Time reference ID is already initialized	No calculation performed	XL_CFI_TIME_REF_INIT_FILE_STATUS_ERR	17
ERR	Could not find out the input file types	No calculation performed	XL_CFI_TIME_REF_INIT_FILE_DETECT_FILE_ERR	18
ERR	The input file type is not correct	No calculation performed	XL_CFI_TIME_REF_INIT_FILE_WRONG_FILE_TYPE_ERR	19
ERR	Input time reference should be UTC for DORIS Navigator files	No calculation performed	XL_CFI_TIME_REF_INIT_FILE_TIME_REF_FOR_DORIS_ERR	20

ERR	Error loading orbit files	No calculation performed	XL_CFI_TIME_REF_INIT_FILE_LOAD_FILES_ERR	21
WARN	Only one OSF file is admitted for this initialisation mode	Calculation performed using first OSV file introduced	XL_CFI_TIME_REF_INIT_FILE_ONLY_FIRST_OSF_WARN	22
WARN	Time init mode option not currently enabled for file	Calculation performed with option XL_SEL_FILE	XL_CFI_TIME_REF_INIT_FILE_INIT_MODE_WARN	23
ERR	Input IERS Bulletins and initialization mode incompatible	No calculation performed	XL_CFI_TIME_REF_INIT_FILE_IERS_INIT_ERR	24
ERR	Input Bulletins B and A are not compatible. The gap or the overlap between the 2 files is not correct	No calculation performed	XL_CFI_TIME_REF_INIT_FILE_IERS_B_A_WRONG_ERR	25
ERR	Could not compute leap second	No calculation performed	XL_CFI_TIME_REF_INIT_FILE_LEAP_SEC_ERR	26

7.2 xl_time_ref_init

7.2.1 Overview

The **xl_time_ref_init** CFI function initializes time correlations between TAI, UTC, UT1 and GPS times from input reference times for time ranges from -18262.0 and +36524.0 decimal days.

A complete calling sequence of the time reference computations is presented in section 4.2.

7.2.2 Calling interface

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

```
#include <explorer_lib.h>
{
      long orbit_num;
      double time[4], anx_time, orbit_duration;
       xl_time_id time_id = {NULL};
      long ierr[XL_NUM_ERR_TIME_REF_INIT], status;


      status =    xl_time_ref_init (time, &orbit_num, &anx_time,
                                    &orbit_duration, &time_id, ierr);
}
```

Note that input time vector must be indexed using the existing enumeration for time references.

The `XL_NUM_ERR_TIME_REF_INIT` constant is defined in the file explorer_lib.h.

7.2.3 Input parameters

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

Table 14: Input parameters of xl_time_ref_init function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
time	double[4]	[0]	TAI input time	Decimal days (Processing format)	[-18262.0,36524.0]
		[1]	UTC input time	Decimal days (Processing format)	[-18262.0,36524.0]
		[2]	UT1 input time	Decimal days (Processing format)	[-18262.0,36524.0]
		[3]	GPS input time	Decimal days	[-18262.0,36524.0]

				(Processing format)	
orbit_num	long*	-	Absolute orbit number at the ref erence time	-	>=0
anx_time	double*	-	Time since Ascending node crossing at the ref erence time	Seconds	[0,orbit_duration]
orbit_duration	double*	-	Duration of the orbit containing the reference time	Seconds	>0

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

- Time vector can be accessible by means of enumeration values, as defined in [GEN_SUM].

7.2.4 Output parameters

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

Table 15: Output parameters of xl_time_ref_init function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_time_ref_init	long	-	Status flag	-	-
time_id	xl_time_id*	-	Structure that contains the time correlations.	-	-
ierr	long	-	Error vector	-	-

7.2.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_time_ref_init** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 by translating the error vector returned by the **xl_time_ref_init** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 16: Error messages of xl_time_ref_init function

Error type	Error message	Cause and impact	Error code	Error No
ERR	TAI time is out of range	No calculation performed	XL_CFI_TIME_REF_INIT_TAI_ERR	0
ERR	UTC time is out of range	No calculation performed	XL_CFI_TIME_REF_INIT_UTC_ERR	1

ERR	UT1 time is out of range	No calculation performed	XL_CFI_TIME_REF_INIT_UT1_ERR	2
ERR	GPS time is out of range	No calculation performed	XL_CFI_TIME_REF_INIT_GPS_ERR	3
ERR	Absolute orbit number is negative	No calculation performed	XL_CFI_TIME_REF_INIT_ORBNUM_ERR	4
ERR	Elapsed time since ANX is negative	No calculation performed	XL_CFI_TIME_REF_INIT_ANXTIME_ERR	5
ERR	Orbit duration is negative	No calculation performed	XL_CFI_TIME_REF_INIT_ORBDUR_ERR	6
ERR	ANX time is bigger than orbit duration	No calculation performed	XL_CFI_TIME_REF_INIT_COMP_ERR	7
ERR	Memory allocation error	No calculation performed	XL_CFI_TIME_REF_INIT_MEMORY_ERR	8
ERR	Time reference ID is already initialized	No calculation performed	XL_CFI_TIME_REF_INIT_STATUS_ERR	9

7.3 xl_time_id_init

7.3.1 Overview

The **xl_time_id_init** CFI function initializes time correlations between TAI, UTC, UT1 and GPS times using any of the following data:
•Set of data read from files (see Table 9 for the allowed file types)
•Data set by the user for the time correlations

7.3.2 Calling interface

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

```
#include <explorer_lib.h>
{
        long time_model, time_init_mode, time_ref;
        long orbit0, orbit1;
        double time0, time1, val_time0, val_time1;
        xl_time_id_init_data **init_data;

        status = xl_time_id_init (&time_model,
                                  &init_data,
                                  &time_init_mode, &time_ref,
                                  &time0, &time1,
                                  &orbit0, &orbit1,
                                  /* output */
                                  &val_time0, &val_time1,
                                  &time_id,
                                  ierr);
}
```

Note that input time vector must be indexed using the existing enumeration for time references.

The `XL_NUM_ERR_TIME_ID_INIT` constant is defined in the file explorer_lib.h.

7.3.3 Input parameters

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

Table 17: Input parameters of xl_time_id_init function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
time_model	long *	-	Time model ID	-	Complete except XL_TIMEMOD_USER
init_data	xl_time_id_init_data *	-	Structure with data for the time initialization	-	-
time_init_mode	long *	-	Flag for selecting the time range of the initialisation: It	-	Select either: · XL_SEL_FILE

			could be the whole file (*XL_SEL_FILE*), the orbit range given by orbit0-orbit1 (*XO_SEL_ORBIT*) or the time range given by time0-time1(*XO_SEL_TIME*)		· XL_SEL_ORBIT · XL_SEL_TIME - XL_SEL_ORBIT is not allowed for IERS Bulletins (any format) nor DORIS Navigator files - XL_SEL_ORBIT and XL_SEL_TIME are not enabled for OSF
time_ref	long *	-	Time reference ID	-	Complete. If the input file is a DORIS Navigator file and the time_init_mode is XL_SEL_TIME, then only time_ref allowed is XL_TIME_UTC.
time0	double*	-	If: *time_init_mode=XL_SEL_TIME* Start of the time range defined by [time0,time1]	Decimal days (Processing format)	[-18262.0,36524.0]
time1	double*	-	If: *time_init_mode=XL_SEL_TIME* End of the time range defined by [time0,time1]	Decimal days (Processing format)	[-18262.0,36524.0] > time0
orbit0	long*	-	If: *time_init_mode=XL_SEL_ORBIT* Absolute orbit number corresponding to the start of the time range defined by [ANX_{orbit0}, $ANX_{orbit1+1}$]	-	>= 0
orbit1	long*	-	If: *time_init_mode=XL_SEL_ORBIT* Absolute orbit number corresponding to the end of the time range defined by [ANX_{orbit0}, $ANX_{orbit1+1}$]	-	>orbit0

7.3.4 Output parameters

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

Table 18: Output parameters of xl_time_id_init function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_time_id_init	long	-	Status flag	-	-
val_time0	double*	-	Validity start time of the initialization (see table 10).	Decimal days (Processing format)	[-18262.0,36524.0]
val_time1	double*	-	Validity end time of	Decimal days	[-18262.0,36524.0]

			the initialization (see table 10).	(Processing format)	
time_id	xl_time_id*	-	Structure with the time correlations.	-	-
lerr	long	-	Error vector	-	-

7.3.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_time_id_init** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 by translating the error vector returned by the **xl_time_id_init** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 19: Error messages of xl_time_id_init function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Time reference ID is already initialized	No calculation performed	XL_CFI_TIME_ID_INIT_STATUS_ERR	0
ERR	Time model ID is not correct	No calculation performed	XL_CFI_TIME_ID_INIT_TIME_MODEL_ERR	1
ERR	Time init mode ID is not correct	No calculation performed	XL_CFI_TIME_ID_INIT_INIT_MODE_ERR	2
ERR	Time reference ID is not correct	No calculation performed	XL_CFI_TIME_ID_INIT_TIME_ERR	3
ERR	Reference start time out of limits	No calculation performed	XL_CFI_TIME_ID_INIT_DAY_0_ERR	4
ERR	Reference end time out of limits	No calculation performed	XL_CFI_TIME_ID_INIT_DAY_1_ERR	5
ERR	Wrong reference time range	No calculation performed	XL_CFI_TIME_ID_INIT_DAY_RANGE_ERR	6
ERR	Reference start orbit is negative	No calculation performed	XL_CFI_TIME_ID_INIT_ORB_0_ERR	7
ERR	Reference end orbit is negative	No calculation performed	XL_CFI_TIME_ID_INIT_ORB_1_ERR	8
ERR	Wrong reference orbit range	No calculation performed	XL_CFI_TIME_ID_INIT_ORB_RANGE_ERR	9
ERR	No data in the input structures	No calculation performed	XL_CFI_TIME_ID_INIT_NO_DATA_ERR	10
ERR	input data structure contains	No calculation performed	XL_CFI_TIME_ID_INIT_I	11

	data for diferent file types		NCONSISTENT_FILES_ERR	
ERR	Memory allocation error	No calculation performed	XL_CFI_TIME_ID_INIT_MEMORY_ERR	12
ERR	No data in the input structure for the requested initialization range	No calculation performed	XL_CFI_TIME_ID_INIT_NO_DATA_IN_RANGE_ERR	13
ERR	Error trying to initialize the time_id	No calculation performed	XL_vTIME_ID_INIT_TIME_CORR_INIT_ERR	14
ERR	Error merging the input set of files	No calculation performed	XL_CFI_TIME_ID_INIT_LOAD_TIME_INIT_LIST_ERR,ERR	15
ERR	Input Time model is inconsistent with the data file type	No calculation performed	XL_CFI_TIME_ID_INIT_WRONG_TIME_MODEL_ERR	16
ERR	Incorrect input data type. It should be XL_FILE_DATA or XL_TIME_CORRELATIONS_DATA	No calculation performed	XL_CFI_TIME_ID_INIT_WRONG_DATA_TYPE_ERR	17
WARN	Time init mode option not enabled for file	Calculation performed All data in OSF range is used for computation	XL_CFI_TIME_ID_INIT_INIT_MODE_WARN	18
WARN	Only one OSF file is admitted for this initialisation mode	Calculation performed with the first OSF data structure in the init_data	XL_CFI_TIME_ID_INIT_ONLY_FIRST_OSF_WARN	19
ERR	Invalid data type or file type or bulletin type. Time mode can not be automatically detected.	No calculation performed	XL_CFI_TIME_ID_INIT_INVALID_FILE_TYPE_ERR	20

7.4 xl_time_close

7.4.1 Overview

The **xl_time_close** CFI function cleans up any memory allocation performed by the initialization functions.

A complete calling sequence of the time reference computations is presented in section 4.2..

7.4.2 Calling interface

The calling interface of the **xl_time_close** CFI function is the following:

```
#include <explorer_lib.h>
{
       xl_time_id time_id = {NULL};
      long ierr[XL_NUM_ERR_TIME_CLOSE], status;
      status =    xl_time_close (&time_id, ierr);
}
```

7.4.3 Input parameters

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

Table 20: Input parameters of xl_time_close function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
time_id	xl_time_id*	-	Structure that contains the time correlations.	-	-

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

- Satellite ID: sat_id. See [GEN_SUM].

7.4.4 Output parameters

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

Table 21: Output parameters of xl_time_close function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_time_close	long	-	Status flag	-	-
ierr	long	-	Error vector	-	-

7.4.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_time_close** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 **xl_time_close** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM])

Table 22: Error messages of xl_time_close function

Error type	Error message	Cause and impact	Error code	Error No
ERR	The Time Id is not initialized or it could be in use by anoter Id.	No calculation performed	XL_CFI_TIME_CLOSE_WRONG_ID_ERR	0

7.5 xl_time_get_id_data

7.5.1 Overview

The **xl_time_get_id_data** CFI function returns a data structure containing the data used for the time initialisation.

7.5.2 Calling interface

The calling interface of the **xl_time_get_id_data** CFI function is the following:

```
#include <explorer_lib.h>
{
        xl_time_id time_id;
        xl_time_id_data data;
       long status;
       status =     xl_time_get_id_data (&time_id, &data);
}
```

7.5.3 Input parameters

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

Table 23: Input parameters of xl_time_get_id_data function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
time_id	xl_time_id*	-	Structure that contains the time correlations.	-	-

7.5.4 Output parameters

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

Table 24: Output parameters of xl_time_get_id_data function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_time_get_id_data	long	-	Status flag	-	-
data	xl_time_id_data	-	Time ID data	-	-

The data structure **xl_time_id_get_id_data** can be seen in Table 8.

7.5.5 Warnings and errors

This function does not return any error/warning code. Only the status of the function indicates if the execution was correct or not.

The possible causes of error are:

- The time_id was not initialised.

7.6 xl_time_set_id_data

7.6.1 Overview

The **xl_time_set_id_data** CFI function changes the time correlations that are stored within a time_id.

7.6.2 Calling interface

The calling interface of the **xl_time_set_id_data** CFI function is the following:

```
#include <explorer_lib.h>
{
        xl_time_id time_id;
        xl_time_id_data data;
       long status;
       status =     xl_time_set_time_id (&time_id, &data);
}
```

7.6.3 Input parameters

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

Table 25: Input parameters of xl_time_set_id_data function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
time_id	xl_time_id*	-	Structure that contains the time correlations (input/outpur parameter)	-	-

7.6.4 Output parameters

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

Table 26: Output parameters of xl_time_set_id_data function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_time_set_id_data	long	-	Status flag	-	-
time_id	xl_time_id*	-	Structure that contains the time correlations (input/outpur	-	-

			parameter)		
data	xl_time_id_data	-	Time ID data	-	-

The data structure **xl_time_set_id_data** can be seen in Table 8.

7.6.5 Warnings and errors

This function does not return any error/warning code. Only the status of the function indicates if the execution was correct or not.

The possible causes of error are:

- The time_id was not initialised.

7.7 xl_run_init

7.7.1 Overview

The **xl_run_init** CFI function groups into a single *id* the *satellite Id*, the *time Id* and the *model_id*, creating a *run Id*.

7.7.2 Calling interface

The calling interface of the **xl_run_init** CFI function is the following:

```
#include <explorer_lib.h>
{
        long sat_id, run_id;
        xl_model_id model_id = {NULL};
        xl_time_id time_id = {NULL};
        long ierr[XL_NUM_ERR_RUN_INIT], status;
        status =     xl_run_init (&sat_id, &time_id, &model_id,
                                  &run_id, ierr);
}
```

7.7.3 Input parameters

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

Table 27: Input parameters of xl_run_init function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
sat_id	long *	-	Satellite ID	-	Complete
time_id	xl_time_id*	-	Structure that contains the time correlations.	-	-
model_id	xl_model_id*	-	Model ID	-	-

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

- Satellite ID: sat_id. See [GEN_SUM].

7.7.4 Output parameters

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

Table 28: Output parameters of xl_run_init function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_run_init	long	-	Status flag	-	-
run_id	long *	-	Run ID	-	>=0
ierr	long	-	Error vector	-	-

7.7.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_run_init** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 **xl_run_init** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM])

Table 29: Error messages of xl_run_init function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Maximum number of initializations reached	No calculation performed	XL_CFI_RUN_INIT_MAX_INIT_ERR	0
ERR	Satellite ID is not correct	No calculation performed	XL_CFI_RUN_INIT_SAT_ERR	1
ERR	Time ID is not initialized	No calculation performed	XL_CFI_RUN_INIT_TIME_INIT_ERR	2
ERR	Memory allocation error	No calculation performed	XL_CFI_RUN_INIT_MEMORY_ERR	3
ERR	Inconsistency between Ids within the run_id	No calculation performed	XL_CFI_RUN_INIT_INCONSISTENCY_ERR	4
ERR	Could not lock other execution threads	No calculation performed	XL_CFI_RUN_INIT_LOCK_ERR	5
ERR	Could not unlock other execution threads	No calculation performed	XL_CFI_RUN_INIT_UNLOCK_ERR	6

7.8 xl_run_get_ids

7.8.1 Overview

The **xl_run_get_ids** CFI function returns the *ids* being used.

7.8.2 Calling interface

The calling interface of the **xl_run_get_ids** CFI function is the following:

```
#include <explorer_lib.h>
{
      long sat_id, run_id;
      xl_time_id  time_id  = {NULL};
      xl_model_id model_id = {NULL};
      xl_run_get_ids (&run_id,
                      &sat_id, &time_id &model_id);
}
```

7.8.3 Input parameters

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

Table 30: Input parameters of xl_run_get_ids function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
run_id	long *	-	Run ID	-	>=0

7.8.4 Output parameters

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

Table 31: Output parameters of xl_run_get_ids function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_run_get_ids	void	-	-	-	-
sat_id	long*	-	Satellite ID	-	-
time_id	xl_time_id*	-	Structure that contains the time correlations.	-	-
model_id	xl_model_id*	-	Model ID	-	-

7.8.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_run_get_ids** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 **xl_run_get_ids** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM])

TBW

7.9 xl_run_close

7.9.1 Overview

The **xl_run_close** CFI function cleans up any memory allocation performed by the initialization functions.

7.9.2 Calling interface

The calling interface of the **xl_run_close** CFI function is the following:

```
#include <explorer_lib.h>
{
      long run_id;
      xl_run_close (&run_id);
}
```

7.9.3 Input parameters

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

Table 32: Input parameters of xl_run_close function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
run_id	long *	-	Run ID	-	>=0

7.9.4 Output parameters

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

Table 33: Output parameters of xl_run_close function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_run_close	void	-	-	-	-

7.9.5 Warnings and errors

No errors have been envisaged for **xl_run_close**.

7.10xl_time_ascii_to_ascii

7.10.1Overview

The **xl_time_ascii_to_ascii** CFI function transforms a time expressed in a given ASCII format and reference (TAI, UTC, UT1 or GPS) into a time in a different ASCII format and/or reference (TAI, UTC, UT1 or GPS).

7.10.2Calling Interface

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

```
#include <explorer_lib.h>
{
      long ascii_id_in, ascii_id_out;
      long time_ref_in, time_ref_out;
      char ascii_in[XL_TIME_ASCII_DIM_MAX];
      char ascii_out[XL_TIME_ASCII_DIM_MAX];
       xl_time_id time_id = {NULL};
      long ierr[XL_NUM_ERR_ASCII_ASCII], status;

      status =    xl_time_ascii_to_ascii(&time_id, &ascii_id_in,
                      &time_ref_in, ascii_in, &ascii_id_out,
                      &time_ref_out, ascii_out, ierr);

      /* Or, using the run_id */
      long run_id;

      status =    xl_time_ascii_to_ascii_run(&run_id, &ascii_id_in,
                      &time_ref_in, ascii_in, &ascii_id_out,
                      &time_ref_out, ascii_out, ierr);
}
```

The `XL_TIME_ASCII_DIM_MAX` and `XL_NUM_ERR_ASCII_ASCII` constants are defined in the file *explorer_lib.h.*

7.10.3Input Parameters

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

Table 34: Input parameters of xl_time_ascii_to_ascii function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
time_id	xl_time_id*	-	Structure that contains the time correlations.	-	-
ascii_id_in	long *	-	ASCII format ID	-	Complete
time_ref_in	long *	-	Time reference ID	-	Complete
ascii_in	char	See Table 4 and Table 5	Time in ASCII for mat	See Table 4 and Table 5	See Table 4 and Table 5
ascii_id_out	long *	-	ASCII format ID	-	Complete
time_ref_out	long *	-	Time reference ID	-	Any except XL_TIME_UND EF

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

- ASCII format ID: ascii_id_in and ascii_id_out. Current document, section 6.2.
- Time reference ID: time_ref_in and time_ref_out. See [GEN_SUM].

It is important to point out the usage of the **time_ref_in** parameter in the frame of the current function:

- If **time_ref_in** input parameter is defined, it shall be used by the function.
- If **time_ref_in** input parameter is undefined, it shall be used the time reference part from the ascii format string. In case this is omitted, an error shall be returned.

Note that for the function to work correctly, the time references should be properly initialised before calling the function (see section 4.2.for details), unless time_ref_in = time_ref_out.

7.10.4Output Parameters

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

Table 35: Output parameters of xl_time_ascii_to_ascii

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_time_ascii_to_ascii	long	-	Status flag	-	-
ascii_out	char	See Table 4 Table 5	Time in ASCII format	See Table 4 and Table 5	See Table 4 and Table 5
ierr	long	-	Error vector	-	-

7.10.5Warnings and Errors

Next table lists the possible error messages that can be returned by the **xl_time_ascii_to_ascii** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 by translating the error vector returned by the **xl_time_ascii_to_ascii** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM])

Table 36: Error messages of xl_time_ascii_to_ascii function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Input ascii format ID is not correct	No calculation performed	XL_CFI_TIME_ASCII_ASCII_ASCII_IN_ERR	0
ERR	Input time reference ID is not correct	No calculation performed	XL_CFI_TIME_ASCII_ASCII_TIME_IN_ERR	1
ERR	Input ascii format is not correct	No calculation performed	XL_CFI_TIME_ASCII_ASCII_FORMAT_IN_ERR	2
ERR	Input time reference inconsistent with the time reference in the date	No calculation performed	XL_CFI_TIME_ASCII_ASCII_REF_INC_IN_ERR	3
ERR	Output ascii format ID is not correct	No calculation performed	XL_CFI_TIME_ASCII_ASCII_ASCII_OUT_ERR	4
ERR	Output time reference ID is not correct	No calculation performed	XL_CFI_TIME_ASCII_ASCII_TIME_OUT_ERR	5
ERR	Input ascii year is out of range	No calculation performed	XL_CFI_TIME_ASCII_ASCII_ASCII_YEAR_IN_ERR	6
ERR	Input ascii month is out of range	No calculation performed	XL_CFI_TIME_ASCII_ASCII_ASCII_MONTH_IN_ERR	7
ERR	Input ascii day is out of range	No calculation performed	XL_CFI_TIME_ASCII_ASCII_ASCII_DAY_IN_ERR	8
ERR	Input ascii hour is out of range	No calculation performed	XL_CFI_TIME_ASCII_ASCII_ASCII_HOUR_IN_ERR	9
ERR	Input ascii minutes are out of range	No calculation performed	XL_CFI_TIME_ASCII_ASCII_ASCII_MIN_IN_ERR	10
ERR	Input ascii seconds are out of range	No calculation performed	XL_CFI_TIME_ASCII_ASCII_ASCII_SEC_IN_ERR	11
ERR	Input ascii microseconds are out of range	No calculation performed	XL_CFI_TIME_ASCII_ASCII_ASCII_MICROSEC_IN_ERR	12
ERR	Internal error: Input Gregorian date to MJD transformation failed	No calculation performed	XL_CFI_TIME_ASCII_ASCII_MJD_IN_ERR	13
ERR	Internal error: Output ascii MJD is out of range	No calculation performed	XL_CFI_TIME_ASCII_ASCII_ASCII_MJD_OUT_ERR	14
ERR	Internal error: Output ascii year is out of range	No calculation performed	XL_CFI_TIME_ASCII_ASCII_ASCII_YEAR_OUT_ERR	15
ERR	Internal error: Output ascii month is out of range	No calculation performed	XL_CFI_TIME_ASCII_ASCII_ASCII_MONTH_OUT_ER R	16
ERR	Internal error: Output ascii day is out of range	No calculation performed	XL_CFI_TIME_ASCII_ASCII_ASCII_DAY_OUT_ERR	17
ERR	Internal error: Output ascii hour is out of range	No calculation performed	XL_CFI_TIME_ASCII_ASCII_ASCII_HOUR_OUT_ERR	18
ERR	Internal error: Output ascii minutes are out of range	No calculation performed	XL_CFI_TIME_ASCII_ASCII_ASCII_MIN_OUT_ERR	19

ERR	Internal error: Output ascii seconds are out of range	No calculation performed	XL_CFI_TIME_ASCII_ASCII_ASCII_SEC_OUT_ERR	20
ERR	Internal error: Output ascii microseconds are out of range	No calculation performed	XL_CFI_TIME_ASCII_ASCII_ASCII_MICROSEC_OUT_ERR	21
ERR	Internal error: Output ascii format is not correct	No calculation performed	XL_CFI_TIME_ASCII_ASCII_FORMAT_OUT_ERR	22
ERR	Time reference not initialised	No calculation performed	XL_CFI_TIME_ASCII_ASCII_REF_INIT_ERR	23
WARN	Time out of initialization range	Calculation performed. A message informs the user.	XL_CFI_TIME_ASCII_ASCII_REF_INIT_WARN	24
WARN	Bulletin A: previous computation performed inside file interval, current performed with formula	Calculation performed. A message informs the user.	XL_CFI_TIME_ASCII_ASCII_BUL_A_FORMULA_WARN	25
WARN	Bulletin B+A: current computation performed inside B-A gap. Previous computation was done inside B or A files intervals.	Calculation performed. A message informs the user.	XL_CFI_TIME_ASCII_ASCII_BUL_B_A_GAP_WARN	26
WARN	Previous computation performed inside initialization validity, current computation performed outside initialization validity	Calculation performed. A message informs the user.	XL_CFI_TIME_ASCII_ASCII_VALIDITY_WARN	32
WARN	EOM detected but not compliant with EO GS File Format Standard	Calculation performed. A message informs the user.	XL_CFI_TIME_ASCII_ASCII_EOM_FFS_COMPLIANCE_WARN	33

7.11xl_time_ascii_to_processing

7.11.1Overview

The **xl_time_ascii_to_processing** CFI function transforms a time expressed in a given ASCII format and reference (TAI, UTC, UT1 or GPS) into a time in Processing format, performing a reference transformation if necessary (to TAI, UTC, UT1 or GPS).

User should be aware that the use of UTC in Processing format is not encouraged, due to the discontinuty that is caused by the introduction of leap seconds. See [IERS] for further details.

7.11.2Calling Interface

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

```
#include <explorer_lib.h>
{
      long ascii_id_in, proc_id_out;
      long time_ref_in, time_ref_out;
      char ascii_in[XL_TIME_ASCII_DIM_MAX];
      double processing_out;
       xl_time_id time_id = {NULL};
      long ierr[XL_NUM_ERR_ASCII_PROC], status;

      status =    xl_time_ascii_to_processing(&time_id, &ascii_id_in,
                      &time_ref_in, ascii_in, &proc_id_out,
                      &time_ref_out, &processing_out, ierr);

      /* Or, using the run_id */
      long run_id;

      status =    xl_time_ascii_to_processing_run(&run_id, &ascii_id_in,
                      &time_ref_in, ascii_in, &proc_id_out,
                      &time_ref_out, &processing_out, ierr);

}
```

The `XL_TIME_ASCII_DIM_MAX` and `XL_NUM_ERR_ASCII_PROC` constants are defined in the file

explorer_lib.h.

7.11.3Input Parameters

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

Table 37: Input parameters of xl_time_ascii_to_processing function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
time_id	xl_time_id*	-	Structure that contains the time correlations.	-	-
ascii_id_in	long *	-	ASCII format ID	-	Complete
time_ref_in	long *	-	Time reference ID	-	Complete
ascii_in	char	See Table 4 and Table 5	Time in ASCII for mat	See Table 4 and Table 5	See Table 4 and Table 5
proc_id_out	long *	-	Processing format ID	-	Complete
time_ref_out	long *	-	Time reference ID	-	Any except XL_TIME_UND EF

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

- ASCII format ID: ascii_id_in. Current document, section 6.2.
- Time reference ID: time_ref_in and time_ref_out. See [GEN_SUM].
- Processing format ID: proc_id_out. Current document, section 6.2

It is important to point out the usage of the **time_ref_in** parameter in the frame of the current function:

- If **time_ref_in** input parameter is defined, it shall be used by the function.
- If **time_ref_in** input parameter is undefined, it shall be used the time reference part from the ascii format string. In case this is omitted, an error shall be returned.

Note that for the function to work correctly, the time references should be properly initialised before calling the function (see section 4.2. for details), unless time_ref_in = time_ref_out.

7.11.4Output Parameters

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

Table 38: Output parameters of xl_time_ascii_to_processing

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_time_ascii_to_pro cessing	long	-	Status flag	-	-
processing_out	double*	-	Time in Processing Format	Decimal days, MJD2000 (Processing)	[-18262.0,36524.0]
ierr	long	-	Error vector	-	-

7.11.5Warnings and Errors

Next table lists the possible error messages that can be returned by the **xl_time_ascii_to_processing** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 by translating the error vector returned by the **xl_time_ascii_to_processing** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM])

Table 39: Error messages of xl_time_ascii_to_processing function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Input ascii format ID is not correct	No calculation performed	XL_CFI_TIME_ASCII_PROC_ASCII_IN_ERR	0
ERR	Input time reference ID is not correct	No calculation performed	XL_CFI_TIME_ASCII_PROC_TIME_IN_ERR	1
ERR	Input format is not correct	No calculation performed	XL_CFI_TIME_ASCII_PROC_FORMAT_IN_ERR	2
ERR	Input time reference inconsistent with the time reference in the date	No calculation performed	XL_CFI_TIME_ASCII_PROC_REF_INC_IN_ERR	
ERR	Output processing format ID is not correct	No calculation performed	XL_CFI_TIME_ASCII_PROC_PROC_OUT_ERR	4
ERR	Output time reference ID is not correct	No calculation performed	XL_CFI_TIME_ASCII_PROC_TIME_OUT_ERR	5
ERR	Year is out of range	No calculation performed	XL_CFI_TIME_ASCII_PROC_YEAR_ERR	6
ERR	Month is out of range	No calculation performed	XL_CFI_TIME_ASCII_PROC_MONTH_ERR	7
ERR	Day is out of range	No calculation performed	XL_CFI_TIME_ASCII_PROC_DAY_ERR	8
ERR	Hour is out of range	No calculation performed	XL_CFI_TIME_ASCII_PROC_HOUR_ERR	9
ERR	Minutes are out of range	No calculation performed	XL_CFI_TIME_ASCII_PROC_MIN_ERR	10
ERR	Seconds are out of range	No calculation performed	XL_CFI_TIME_ASCII_PROC_SEC_ERR	11
ERR	Microseconds are out of range	No calculation performed	XL_CFI_TIME_ASCII_PROC_MICROSEC_ERR	12
ERR	Internal Error: Input Gregorian date to MJD transformation failed	No calculation performed	XL_CFI_TIME_ASCII_PROC_MJD_ERR	13
ERR	Time reference not initialised	No calculation performed	XL_CFI_TIME_ASCII_PROC_REF_INIT_ERR	14
WARN	Time out of initialization range	Calculation performed. A message informs the user.	XL_CFI_TIME_ASCII_PROC_REF_INIT_WARN	15
WARN	Bulletin A: previous computation performed inside file interval, current performed with formula	Calculation performed. A message informs the user.	XL_CFI_TIME_ASCII_PROC_BUL_A_FORMULA_WARN	16

WARN	Bulletin B+A: current computation performed inside B-A gap. Previous computation was done inside B or A files intervals.	Calculation performed. A message informs the user.	XL_CFI_TIME_ASCII_PROC_BUL_B_A_GAP_WARN	17
WARN	Previous computation performed inside initialization validity, current computation performed outside initialization validity	Calculation performed. A message informs the user.	XL_CFI_TIME_ASCII_PROC_VALIDITY_WARN	18

7.12xl_time_ascii_to_transport

7.12.1Overview

The **xl_time_ascii_to_transport** CFI function transforms a time expressed in a given ASCII format and reference (TAI, UTC, UT1 or GPS) into a time in a Transport format, performing a reference transformation if necessary (to TAI, UTC, UT1 or GPS).

7.12.2Calling Interface

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

```
#include <explorer_lib.h>
{
      long ascii_id_in, trans_id_out;
      long time_ref_in, time_ref_out;
      char ascii_in[XL_TIME_ASCII_DIM_MAX];
      long transport_out[XL_TIME_TRANS_DIM_MAX];
       xl_time_id time_id = {NULL};
      long ierr[XL_NUM_ERR_ASCII_TRANS], status;

      status =    xl_time_ascii_to_transport(&time_id, &ascii_id_in,
                      &time_ref_in, ascii_in, &trans_id_out,
                      &time_ref_out, transport_out, ierr);

      /* Or, using the run_id */
      long run_id;

      status =    xl_time_ascii_to_transport_run(&run_id, &ascii_id_in,
                      &time_ref_in, ascii_in, &trans_id_out,
                      &time_ref_out, transport_out, ierr);

}
```

The `XL_TIME_TRANS_DIM_MAX`, `XL_TIME_ASCII_DIM_MAX`, `XL_NUM_ERR_ASCII_TRANS` constants are defined in the file *explorer_lib.h*.

7.12.3Input Parameters

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

Table 40: Input parameters of xl_time_ascii_to_transport function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
time_id	xl_time_id*	-	Structure that contains the time correlations.	-	-
ascii_id_in	long *	-	ASCII format ID	-	Complete
time_ref_in	long *	-	Time reference ID	-	Complete
ascii_in	char	See Table 4 and Table 5	Time in ASCII for mat	See Table 4 and Table 5	See Table 4 and Table 5
trans_id_out	long *	-	Transport format ID	-	Complete
time_ref_out	long *	-	Time reference ID	-	Any except XL_TIME_UND EF

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

- Satellite ID: sat_id. See [GEN_SUM].
- ASCII format ID: trans_id_in. Current document, section 6.2.
- Time reference ID: time_ref_in and time_ref_out. See [GEN_SUM].
- Transport format ID: trans_id_out. Current document, section 6.2.

It is important to point out the usage of the time_ref_in parameter in the frame of the current function:

- If time_ref_in input parameter is defined, it shall be used by the function.
- If time_ref_in input parameter is undefined, it shall be used the time reference part from the ascii format string. In case this is omitted, an error shall be returned.

Note that for the function to work correctly, the time references should be properly initialised before calling the function (see section 4.2. for details), unless time_ref_in = time_ref_out.

7.12.4Output Parameters

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

Table 41: Output parameters of xl_time_ascii_to_transport

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_time_ascii_to_tran sport	long	-	Status flag	-	-
transport_out[dim]	long	See Table 3	Time in Transport for mat	See Table 3	See Table 3
ierr	long	-	Error vector	-	-

7.12.5Warnings and Errors

Next table lists the possible error messages that can be returned by the **xl_time_ascii_to_transport** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 by translating the error vector returned by the **xl_time_ascii_to_transport** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM])

Table 42: Error messages of xl_time_ascii_to_transport function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Input ascii format ID is not correct	No calculation performed	XL_CFI_TIME_ASCII_TRANS_ASCII_IN_ERR	0
ERR	Input time reference ID is not correct	No calculation performed	XL_CFI_TIME_ASCII_TRANS_TIME_IN_ERR	1
ERR	Input format is not correct	No calculation performed	XL_CFI_TIME_ASCII_TRANS_FORMAT_IN_ERR	2
ERR	Input time reference inconsistent with the time reference in the date	No calculation performed	XL_CFI_TIME_ASCII_TRANS_REF_INC_IN_ERR	3
ERR	Output transport format ID is not correct	No calculation performed	XL_CFI_TIME_ASCII_TRANS_TRANS_OUT_ERR	4
ERR	Output time reference ID is not correct	No calculation performed	XL_CFI_TIME_ASCII_TRANS_TIME_OUT_ERR5	5
ERR	Year is out of range	No calculation performed	XL_CFI_TIME_ASCII_TRANS_YEAR_ERR	6
ERR	Month is out of range	No calculation performed	XL_CFI_TIME_ASCII_TRANS_MONTH_ERR	7
ERR	Day is out of range	No calculation performed	XL_CFI_TIME_ASCII_TRANS_DAY_ERR	8
ERR	Hour is out of range	No calculation performed	XL_CFI_TIME_ASCII_TRANS_HOUR_ERR	9
ERR	Minutes are out of range	No calculation performed	XL_CFI_TIME_ASCII_TRANS_MIN_ERR	10
ERR	Seconds are out of range	No calculation performed	XL_CFI_TIME_ASCII_TRANS_SEC_ERR	11
ERR	Microseconds are out of range	No calculation performed	XL_CFI_TIME_ASCII_TRANS_MICROSEC_ERR	12
ERR	Internal Error: Input Gregorian date to MJD transformation failed	No calculation performed	XL_CFI_TIME_ASCII_TRANS_MJD_ERR	13
ERR	Time reference not initialised	No calculation performed	XL_CFI_TIME_ASCII_TRANS_REF_INIT_ERR	14
WARN	Time out of initialization range	Calculation performed. A message informs the user.	XL_CFI_TIME_ASCII_TRANS_REF_INIT_WARN	15
WARN	Bulletin A: previous computation performed inside	Calculation performed. A message informs the user.	XL_CFI_TIME_ASCII_TRANS_BUL_A_FORMULA_WAR	16

	file interval, current performed with formula		N	
WARN	Bulletin B+A: current computation performed inside B-A gap. Previous computation was done inside B or A files intervals.	Calculation performed. A message informs the user.	XL_CFI_TIME_ASCII_TRANS_BUL_B_A_GAP_WARN	17
WARN	Previous computation performed inside initialization validity, current computation performed outside initialization validity	Calculation performed. A message informs the user.	XL_CFI_TIME_ASCII_TRANS_VALIDITY_WARN	18

7.13xl_time_processing_to_ascii

7.13.1Overview

The **xl_time_processing_to_ascii** CFI function transforms a time expressed in Processing format and a given reference (TAI, UTC, UT1 or GPS) into a time in an ASCII format, performing a reference transformation if necessary (to TAI, UTC, UT1 or GPS).

User should be aware that the use of UTC in Processing format is not encouraged, due to the discontinuty that is caused by the introduction of leap seconds. See [IERS] for further details.

7.13.2Calling Interface

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

```
#include <explorer_lib.h>
{
      long proc_id_in, ascii_id_out;
      long time_ref_in, time_ref_out;
      double processing_in;
      char ascii_out[XL_TIME_ASCII_DIM_MAX];
       xl_time_id time_id = {NULL};
      long ierr[XL_NUM_ERR_PROC_ASCII], status;

      status =    xl_time_processing_to_ascii(&time_id, &proc_id_in,
                      &time_ref_in, &processing_in, &ascii_id_out,
                      &time_ref_out, ascii_out, ierr);

      /* Or, using the run_id */
      long run_id;

      status =    xl_time_processing_to_ascii_run(&run_id, &proc_id_in,
                      &time_ref_in, &processing_in, &ascii_id_out,
                      &time_ref_out, ascii_out, ierr);
}
```

7.13.3Input Parameters

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

Table 43: Input parameters of xl_time_processing_to_ascii function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
time_id	xl_time_id*	-	Structure that contains the time correlations.	-	-
proc_id_in	long *	-	Processing format ID	-	Complete
time_ref_in	long *	-	Time reference ID	-	Any except XL_TIME_UNDEF
processing_in	double*	-	Time in Process ing Format	Decimal days, MJD2000 (Processing)	[-18262.0,36524.0]
ascii_id_out	long *	-	ASCII format ID	-	Complete
time_ref_out	long *	-	Time reference ID	-	Any except XL_TIME_UNDEF

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

- Processing format ID: proc_id_in. Current document, section 6.2.
- Time reference ID: time_ref_in and time_ref_out. See [GEN_SUM].
- ASCII format ID: ascii_id_out. Current document, section 6.2.

Note that for the function to work correctly, the time references should be properly initialised before calling the function (see section 4.2. for details), unless time_ref_in = time_ref_out.

7.13.4Output Parameters

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

Table 44: Output parameters of xl_time_processing_to_ascii

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_time_processing_to_ascii	long	-	Status flag	-	-
ascii_out	char	See Table 4 and Table 5	Time in ASCII format	See Table 4 and Table 5	See Table 4 and Table 5
ierr	long	-	Error vector	-	-

7.13.5Warnings and Errors

Next table lists the possible error messages that can be returned by the **xl_time_processing_to_ascii** CFI function after translating the returned error vector into the equivalent list of error messages by calling the

function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 by translating the error vector returned by the **xl_time_processing_to_ascii** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM])

Table 45: Error messages of xl_time_processing_to_ascii function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Input processing format ID is not correct	No calculation performed	XL_CFI_TIME_PROC_ASCII_PROC_IN_ERR	0
ERR	Input time reference ID is not correct	No calculation performed	XL_CFI_TIME_PROC_ASCII_TIME_IN_ERR	1
ERR	Input days out of range	No calculation performed	XL_CFI_TIME_PROC_ASCII_DAY_ERR	2
ERR	Output ascii format ID is not correct	No calculation performed	XL_CFI_TIME_PROC_ASCII_ASCII_OUT_ERR	3
ERR	Output time reference ID is not correct	No calculation performed	XL_CFI_TIME_PROC_ASCII_TIME_OUT_ERR	4
ERR	Internal error: Output ascii MJD is out of range	No calculation performed	XL_CFI_TIME_PROC_ASCII_ASCII_MJD_ERR	5
ERR	Internal error: Output ascii year is out of range	No calculation performed	XL_CFI_TIME_PROC_ASCII_ASCII_YEAR_ERR	6
ERR	Internal error: Output ascii month is out of range	No calculation performed	XL_CFI_TIME_PROC_ASCII_ASCII_MONTH_ERR	7
ERR	Internal error: Output ascii day is out of range	No calculation performed	XL_CFI_TIME_PROC_ASCII_ASCII_DAY_ERR	8
ERR	Internal error: Output ascii hour is out of range	No calculation performed	XL_CFI_TIME_PROC_ASCII_ASCII_HOUR_ERR	9
ERR	Internal error: Output ascii minutes are out of range	No calculation performed	XL_CFI_TIME_PROC_ASCII_ASCII_MIN_ERR	10
ERR	Internal error: Output ascii seconds are out of range	No calculation performed	XL_CFI_TIME_PROC_ASCII_ASCII_SEC_ERR	11
ERR	Internal error: Output ascii microseconds are out of range	No calculation performed	XL_CFI_TIME_PROC_ASCII_ASCII_MICROSEC_ERR	12
ERR	Internal error: Output ascii format is not correct	No calculation performed	XL_CFI_TIME_PROC_ASCII_FORMAT_OUT_ERR	13
ERR	Time reference not initialised	No calculation performed	XL_CFI_TIME_PROC_ASCII_REF_INIT_ERR	14
WARN	Time out of initialization range	Calculation performed. A message informs the user.	XL_CFI_TIME_PROC_ASCII_REF_INIT_WARN	15
WARN	Bulletin A: previous computation perfromed inside file interval, current performed with formula	Calculation performed. A message informs the user.	XL_CFI_TIME_PROC_ASCII_BUL_A_FORMULA_WARN	16
WARN	Bulletin B+A: current computation performed inside B-A gap. Previous computation was done inside B or A files intervals.	Calculation performed. A message informs the user.	XL_CFI_TIME_PROC_ASCII_BUL_B_A_GAP_WARN	17
WARN	Previous computation	Calculation performed.	XL_CFI_TIME_PROC_ASCII	18

	performed inside initialization validity, current computation performed outside initialization validity	A message informs the user.	_VALIDITY_WARN	

7.14xl_time_processing_to_processing

7.14.1Overview

The **xl_time_processing_to_processing** CFI function transforms a time expressed in Processing format and a given reference (TAI, UTC, UT1 or GPS) into a time in Processing format with a different reference (TAI, UTC, UT1 or GPS).

User should be aware that the use of UTC in Processing format is not encouraged, due to the discontinuty that is caused by the introduction of leap seconds. See [IERS] for further details.

7.14.2Calling Interface

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

```
#include <explorer_lib.h>
{
      long proc_id_in, proc_id_out;
      long time_ref_in, time_ref_out;
      double processing_in, processing_out;
       xl_time_id time_id = {NULL};
      long ierr[XL_NUM_ERR_PROC_PROC], status;


      status =    xl_time_processing_to_processing(&time_id,
&proc_id_in,
                        &time_ref_in, &processing_in, &proc_id_out,
                        &time_ref_out, &processing_out, ierr);


      /* Or, using the run_id */
      long run_id;


   status =    xl_time_processing_to_processing_run(&run_id, &proc_id_in,
                        &time_ref_in, &processing_in, &proc_id_out,
                        &time_ref_out, &processing_out, ierr);
}
```

The `XL_NUM_ERR_PROC_PROC` constant is defined in the file *explorer_lib.h.*

7.14.3Input Parameters

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

Table 46: Input parameters of xl_time_processing_to_processing function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
time_id	xl_time_id*	-	Structure that contains the time correlations.	-	-
proc_id_in	long *	-	Processing format ID	-	Complete
time_ref_in	long *	-	Time reference ID	-	Any except XL_TIME_UNDEF
processing_in	double*	-	Time in Process ing Format	Decimal days, MJD2000 (Processing)	[-18262.0,36524.0]
proc_id_out	long *	-	Processing format ID	-	Complete
time_ref_out	long *	-	Time reference ID	-	Any except XL_TIME_UNDEF

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

- Processing format ID: proc_id_in and proc_id_out. Current document, section 6.2.
- Time reference ID: time_ref_in and time_ref_out. See [GEN_SUM].

Note that for the function to work correctly, the time references should be properly initialised before calling the function (see section 4.2. for details), unless time_ref_in = time_ref_out.

7.14.4Output Parameters

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

Table 47: Output parameters of xl_time_processing_to_processing

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_time_processing_to_processing	long	-	Status flag	-	-
processing_out	double*	-	Time in Processing Format	Decimal days, MJD2000 (Processing)	[-18262.0,36524.0]
ierr	long	-	Error vector	-	-

7.14.5Warnings and Errors

Next table lists the possible error messages that can be returned by the **xl_time_processing_to_processing** CFI function after translating the returned error vector into the equivalent list of error messages by calling

the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 by translating the error vector returned by the **xl_time_processing_to_processing** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM])

Table 48: Error messages of xl_time_processing_to_processing function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Input processing format ID is not correct	No calculation performed	XL_CFI_TIME_PROC_PROC_PROC_IN_ERR	0
ERR	Input time reference ID is not correct	No calculation performed	XL_CFI_TIME_PROC_PROC_TIME_IN_ERR	1
ERR	Output processing format ID is not correct	No calculation performed	XL_CFI_TIME_PROC_PROC_PROC_OUT_ERR	2
ERR	Output time reference ID is not correct	No calculation performed	XL_CFI_TIME_PROC_PROC_TIME_OUT_ERR	3
ERR	Number of days out of range	No calculation performed	XL_CFI_TIME_PROC_PROC_DAY_ERR	4
ERR	Time reference not initialised	No calculation performed	XL_CFI_TIME_PROC_PROC_REF_INIT_ERR	5
WARN	Time out of initialization range	Calculation performed. A message informs the user.	XL_CFI_TIME_PROC_PROC_REF_INIT_WARN	6
WARN	Bulletin A: previous computation perfromed inside file interval, current performed with formula	Calculation performed. A message informs the user.	XL_CFI_TIME_PROC_PROC_BUL_A_FORMULA_WARN	7
WARN	Bulletin B+A: current computation performed inside B-A gap. Previous computation was done inside B or A files intervals.	Calculation performed. A message informs the user.	XL_CFI_TIME_PROC_PROC_BUL_B_A_GAP_WARN	8
WARN	Previous computation performed inside initialization validity, current computation performed outside initialization validity	Calculation performed. A message informs the user.	XL_CFI_TIME_PROC_PROC_VALIDITY_WARN	9

7.15xl_time_processing_to_transport

7.15.1Overview

The **xl_time_processing_to_transport** CFI function transforms a time expressed in Processing format and a given reference (TAI, UTC, UT1 or GPS) into a time in a Transport format, performing a reference transformation if necessary (to TAI, UTC, UT1 or GPS).

User should be aware that the use of UTC in Processing format is not encouraged, due to the discontinuty that is caused by the introduction of leap seconds. See [IERS] for further details.

7.15.2Calling Interface

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

```
#include <explorer_lib.h>
{
      long proc_id_in, trans_id_out;
      long time_ref_in, time_ref_out;
      double processing_in;
      long transport_out[XL_TIME_TRANS_DIM_MAX];
       xl_time_id time_id = {NULL};
      long ierr[XL_NUM_ERR_PROC_TRANS], status;

      status =    xl_time_processing_to_transport(&time_id, &proc_id_in,
                      &time_ref_in, &processing_in, &trans_id_out,
                      &time_ref_out, transport_out, ierr);

      /* Or, using the run_id */
      long run_id;

    status =  xl_time_processing_to_transport_run(&run_id, &proc_id_in,
                      &time_ref_in, &processing_in, &trans_id_out,
                      &time_ref_out, transport_out, ierr);
}
```

The `XL_TIME_TRANS_DIM_MAX` and `XL_NUM_ERR_PROC_TRANS` constants are defined in the file *explorer_lib.h.*

7.15.3Input Parameters

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

Table 49: Input parameters of xl_time_processing_to_transport function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
time_id	xl_time_id*	-	Structure that contains the time correlations.	-	-
proc_id_in	long *	-	Processing format ID	-	Complete
time_ref_in	long *	-	Time reference ID	-	Any except XL_TIME_UNDEF
processing_in	double*	-	Time in Process ing Format	Decimal days, MJD2000 (Processing)	[-18262.0,36524.0]
trans_id_out	long *	-	Transport format ID	-	Complete
time_ref_out	long *	-	Time reference ID	-	Any except XL_TIME_UNDEF

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

- Processing format ID: proc_id_in. Current document, section 6.2.
- Time reference ID: time_ref_in and time_ref_out. See [GEN_SUM].
- Transport format ID: trans_id_out. Current document, section 6.2.

Note that for the function to work correctly, the time references should be properly initialised before calling the function (see section 4.2. for details), unless time_ref_in = time_ref_out.

7.15.4Output Parameters

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

Table 50: Output parameters of xl_time_processing_to_transport

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_time_processing_to_transport	long	-	Status flag	-	-
transport_out[dim]	long	See Table 3	Time in Transport for mat	See Table 3	See Table 3
ierr	long	-	Error vector	-	-

7.15.5Warnings and Errors

Next table lists the possible error messages that can be returned by the **xl_time_processing_to_transport** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 error vector returned by the **xl_time_processing_to_transport** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM])

Table 51: Error messages of xl_time_processing_to_transport function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Input processing format ID is not correct	No calculation performed	XL_CFI_TIME_PROC_TRANS_PROC_IN_ERR	0
ERR	Input time reference ID is not correct	No calculation performed	XL_CFI_TIME_PROC_TRANS_TIME_IN_ERR	1
ERR	Output transport format ID is not correct	No calculation performed	XL_CFI_TIME_PROC_TRANS_TRANS_OUT_ERR	2
ERR	Output time reference ID is not correct	No calculation performed	XL_CFI_TIME_PROC_TRANS_TIME_OUT_ERR	3
ERR	Number of days out of range	No calculation performed	XL_CFI_TIME_PROC_TRANS_DAY_ERR	4
ERR	Time reference not initialised	No calculation performed	XL_CFI_TIME_PROC_TRANS_REF_INIT_ERR	5
WARN	Time out of initialization range	Calculation performed. A message informs the user.	XL_CFI_TIME_PROC_TRANS_REF_INIT_WARN	6
WARN	Bulletin A: previous computation perfromed inside file interval, current performed with formula	Calculation performed. A message informs the user.	XL_CFI_TIME_PROC_TRANS_BUL_A_FORMULA_WARN	7
WARN	Bulletin B+A: current computation performed inside B-A gap. Previous computation was done inside B or A files intervals.	Calculation performed. A message informs the user.	XL_CFI_TIME_PROC_TRANS_BUL_B_A_GAP_WARN	8
WARN	Previous computation performed inside initialization validity, current computation performed outside initialization validity	Calculation performed. A message informs the user.	XL_CFI_TIME_PROC_TRANS_VALIDITY_WARN	9

7.16xl_time_transport_to_ascii

7.16.1Overview

The **xl_time_transport_to_ascii** CFI function transforms a time expressed in a given Transport format and reference (TAI, UTC, UT1 or GPS) into a time in an ASCII format, performing a reference transformation if necessary (to TAI, UTC, UT1 or GPS).

7.16.2Calling Interface

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

```
#include <explorer_lib.h>
{
      long trans_id_in, ascii_id_out;
      long time_ref_in, time_ref_out;
      long transport_in[XL_TIME_TRANS_DIM_MAX];
      char ascii_out[XL_TIME_ASCII_DIM_MAX];
       xl_time_id time_id = {NULL};
      long ierr[XL_NUM_ERR_TRANS_ASCII], status;

      status =    xl_time_transport_to_ascii(&time_id, &trans_id_in,
                      &time_ref_in, transport_in, &ascii_id_out,
                      &time_ref_out, ascii_out, ierr);

      /* Or, using the run_id */
      long run_id;

      status =    xl_time_transport_to_ascii_run(&run_id, &trans_id_in,
                      &time_ref_in, transport_in, &ascii_id_out,
                      &time_ref_out, ascii_out, ierr);
}
```

The `XL_TIME_TRANS_DIM_MAX`, `XL_TIME_ASCII_DIM_MAX`, `XL_NUM_ERR_TRANS_ASCII` constants are defined in the file *explorer_lib.h*.

7.16.3Input Parameters

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

Table 52: Input parameters of xl_time_transport_to_ascii function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
time_id	xl_time_id*	-	Structure that contains the time correlations.	-	-
trans_id_in	long *	-	Transport format ID	-	Complete
time_ref_in	long *	-	Time reference ID	-	Any except XL_TIME_UNDEF
transport_in[dim]	long	See tTable 3	Time in Transport format	See Table 3	See Table 3
ascii_id_out	long *	-	ASCII format ID	-	Complete
time_ref_out	long *	-	Time reference ID	-	Any except XL_TIME_UNDEF

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

- Transport format ID: trans_id_in. Current document, section 6.2.
- Time reference ID: time_ref_in and time_ref_out. See [GEN_SUM].
- ASCII format ID: ascii_id_out. Current document, section 6.2.

It is important to point out the usage of the time_ref_out parameter within the current function:

- If the time reference flag for the output is undefined, an error shall be returned.

Note that for the function to work correctly, the time references should be properly initialised before calling the function (see section 4.2. for details), unless time_ref_in = time_ref_out.

7.16.4Output Parameters

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

Table 53: Output parameters of xl_time_transport_to_ascii

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_time_transport_to_ascii	long	-	Status flag	-	-
ascii_out	char	See Table 4 Table 5	Time in ASCII format	See Table 4 and Table 5	See Table 4 and Table 5
ierr	long	-	Error vector	-	-

7.16.5Warnings and Errors

Next table lists the possible error messages that can be returned by the **xl_time_transport_to_ascii** CFI

function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 by translating the error vector returned by the **xl_time_transport_to_ascii** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 54: Error messages of xl_time_transport_to_ascii function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Input transport format ID is not correct	No calculation performed	XL_CFI_TIME_TRANS_ASCII_TRANS_IN_ERR	0
ERR	Input time reference ID is not correct	No calculation performed	XL_CFI_TIME_TRANS_ASCII_TIME_IN_ERR	1
ERR	Number of days out of range	No calculation performed	XL_CFI_TIME_TRANS_ASCII_DAY_ERR	2
ERR	Number of seconds out of range	No calculation performed	XL_CFI_TIME_TRANS_ASCII_SEC_ERR	3
ERR	Number of milliseconds out of range	No calculation performed	XL_CFI_TIME_TRANS_ASCII_MILLISEC_ERR	4
ERR	Number of microseconds out of range	No calculation performed	XL_CFI_TIME_TRANS_ASCII_MICROSEC_ERR	5
ERR	Number of SIRAL extra counter ticks out of range	No calculation performed	XL_CFI_TIME_TRANS_ASCII_TICK_ERR	6
ERR	Output ascii format ID is not correct	No calculation performed	XL_CFI_TIME_TRANS_ASCII_ASCII_OUT_ERR	7
ERR	Output time reference ID is not correct	No calculation performed	XL_CFI_TIME_TRANS_ASCII_TIME_OUT_ERR	8
ERR	Internal error: Output ascii MJD is out of range	No calculation performed	XL_CFI_TIME_TRANS_ASCII_ASCII_MJD_ERR	9
ERR	Internal error: Output ascii year is out of range	No calculation performed	XL_CFI_TIME_TRANS_ASCII_ASCII_YEAR_ERR	10
ERR	Internal error: Output ascii month is out of range	No calculation performed	XL_CFI_TIME_TRANS_ASCII_ASCII_MONTH_ERR	11
ERR	Internal error: Output ascii day is out of range	No calculation performed	XL_CFI_TIME_TRANS_ASCII_ASCII_DAY_ERR	12
ERR	Internal error: Output ascii hour is out of range	No calculation performed	XL_CFI_TIME_TRANS_ASCII_ASCII_HOUR_ERR	13
ERR	Internal error: Output ascii minutes are out of range	No calculation performed	XL_CFI_TIME_TRANS_ASCII_ASCII_MIN_ERR	14
ERR	Internal error: Output ascii seconds are out of range	No calculation performed	XL_CFI_TIME_TRANS_ASCII_ASCII_SEC_ERR	15
ERR	Internal error: Output ascii microseconds are out of range	No calculation performed	XL_CFI_TIME_TRANS_ASCII_ASCII_MICROSEC_ERR	16
ERR	Internal error: Output ascii format is not correct	No calculation performed	XL_CFI_TIME_TRANS_ASCII_FORMAT_OUT_ERR	17
ERR	Time reference not initialised	No calculation performed	XL_CFI_TIME_TRANS_ASCII_REF_INIT_ERR	18
WARN	Time out of initialization range	Calculation performed. A message informs the user.	XL_CFI_TIME_TRANS_ASCII_REF_INIT_WARN	19

WARN	Bulletin A: previous computation perfromed inside file interval, current performed with formula	Calculation performed. A message informs the user.	XL_CFI_TIME_TRANS_ASCII_BUL_A_FORMULA_WARN	20
WARN	Bulletin B+A: current computation performed inside B-A gap. Previous computation was done inside B or A files intervals.	Calculation performed. A message informs the user.	XL_CFI_TIME_TRANS_ASCII_BUL_B_A_GAP_WARN	21
WARN	Previous computation performed inside initialization validity, current computation performed outside initialization validity	Calculation performed. A message informs the user.	XL_CFI_TIME_TRANS_ASCII_VALIDITY_WARN	22

7.17xl_time_transport_to_processing

7.17.1Overview

The **xl_time_transport_to_processing** CFI function transforms a time expressed in a given Transport format and reference (TAI, UTC, UT1 or GPS) into a time in Processing format, performing a reference transformation if necessary (to TAI, UTC, UT1 or GPS).

User should be aware that the use of UTC in Processing format is not encouraged, due to the discontinuity that is caused by the introduction of leap seconds. See [IERS] for further details.

7.17.2Calling Interface

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

```
#include <explorer_lib.h>
{
      long trans_id_in, proc_id_out;
      long time_ref_in, time_ref_out;
      long transport_in[XL_TIME_TRANS_DIM_MAX];
      double processing_out;
       xl_time_id time_id = {NULL};
      long ierr[XL_NUM_ERR_TRANS_PROC], status;


      status =    xl_time_transport_to_processing(&time_id,
&trans_id_in,
                        &time_ref_in, transport_in, &proc_id_out,
                        &time_ref_out, &processing_out, ierr);


      /* Or, using the run_id */
      long run_id;


   status =   xl_time_transport_to_processing_run(&run_id, &trans_id_in,
                        &time_ref_in, transport_in, &proc_id_out,
                        &time_ref_out, &processing_out, ierr);
}
```

The `XL_TIME_TRANS_DIM_MAX` and `XL_NUM_ERR_TRANS_PROC` constants are defined in the file *explorer_lib.h*.

7.17.3Input Parameters

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

Table 55: Input parameters of xl_time_transport_to_processing function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
time_id	xl_time_id*	-	Structure that contains the time correlations.	-	-
trans_id_in	long *	-	Transport format ID	-	Complete
time_ref_in	long *	-	Time reference ID	-	Any except XL_TIME_UNDEF
transport_in[dim]	long	See Table 3	Time in Transport format	See Table 3	See Table 3
proc_id_out	long *	-	Processing format ID	-	Complete
time_ref_out	long *	-	Time reference ID	-	Any except XL_TIME_UNDEF

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

- Transport format ID: trans_id_in. Current document, section 6.2.
- Time reference ID: time_ref_in and time_ref_out. See [GEN_SUM].
- Processing format ID: proc_id_out. Current document, section 6.2

Note that for the function to work correctly, the time references should be properly initialised before calling the function (see section 4.2. for details), unless time_ref_in = time_ref_out.

7.17.4Output Parameters

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

Table 56: Output parameters of xl_time_transport_to_processing

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_time_transport_to_processing	long	-	Status flag	-	-
processing_out	double*	-	Time in Processing Format	Decimal days, MJD2000 (Processing)	[-18262.0,36524.0]
ierr	long	-	Error vector	-	-

7.17.5Warnings and Errors

Next table lists the possible error messages that can be returned by the **xl_time_transport_to_processing** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 by translating the error vector returned by the **xl_time_transport_to_processing** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM])

Table 57: Error messages of xl_time_transport_to_processing function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Input transport format ID is not correct	No calculation performed	XL_CFI_TIME_TRANS_PROC_TRANS_IN_ERR	0
ERR	Input time reference ID is not correct	No calculation performed	XL_CFI_TIME_TRANS_PROC_TIME_IN_ERR	1
ERR	Output processing format ID is not correct	No calculation performed	XL_CFI_TIME_TRANS_PROC_PROC_OUT_ERR	2
ERR	Output time reference ID is not correct	No calculation performed	XL_CFI_TIME_TRANS_PROC_TIME_OUT_ERR	3
ERR	Number of days out of range	No calculation performed	XL_CFI_TIME_TRANS_PROC_DAY_ERR	4
ERR	Number of seconds out of range	No calculation performed	XL_CFI_TIME_TRANS_PROC_SEC_ERR	5
ERR	Number of milliseconds out of range	No calculation performed	XL_CFI_TIME_TRANS_PROC_MILLISEC_ERR	6
ERR	Number of microseconds out of range	No calculation performed	XL_CFI_TIME_TRANS_PROC_MICROSEC_ERR	7
ERR	Number of SIRAL extra counter ticks out of range	No calculation performed	XL_CFI_TIME_TRANS_PROC_TICK_ERR	8
ERR	Time reference not initialised	No calculation performed	XL_CFI_TIME_TRANS_PROC_REF_INIT_ERR	9
WARN	Time out of initialization range	Calculation performed. A message informs the user.	XL_CFI_TIME_TRANS_PROC_REF_INIT_WARN	10
WARN	Bulletin A: previous computation perfromed inside file interval, current performed with formula	Calculation performed. A message informs the user.	XL_CFI_TIME_TRANS_PROC_BUL_A_FORMULA_WARN	11
WARN	Bulletin B+A: current computation performed inside B-A gap. Previous computation was done inside B or A files intervals.	Calculation performed. A message informs the user.	XL_CFI_TIME_TRANS_PROC_BUL_B_A_GAP_WARN	12
WARN	Previous computation performed inside initialization validity, current computation performed outside initialization validity	Calculation performed. A message informs the user.	XL_CFI_TIME_TRANS_PROC_VALIDITY_WARN	13

7.18xl_time_transport_to_transport

7.18.1Overview

The **xl_time_transport_to_transport** CFI function transforms a time expressed in a given Transport format and reference (TAI, UTC, UT1 or GPS) into a time in a different Transport format and/or reference (TAI, UTC, UT1 or GPS).

7.18.2Calling Interface

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

```
#include <explorer_lib.h>
{
      long trans_id_in, trans_id_out;
      long time_ref_in, time_ref_out;
      long transport_in[XL_TIME_TRANS_DIM_MAX];
      long transport_out[XL_TIME_TRANS_DIM_MAX];
      xl_time_id time_id = {NULL};
      long ierr[XL_NUM_ERR_TRANS_TRANS], status;

      status =    xl_time_transport_to_transport(&time_id, &trans_id_in,
                      &time_ref_in, transport_in, &trans_id_out,
                      &time_ref_out, transport_out, ierr);

      /* Or, using the run_id */
      long run_id;

    status =  xl_time_transport_to_transport_run(&run_id, &trans_id_in,
                      &time_ref_in, transport_in, &trans_id_out,
                      &time_ref_out, transport_out, ierr);
}
```

7.18.3Input Parameters

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

Table 58: Input parameters of xl_time_transport_to_transport function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
time_id	xl_time_id*	-	Structure that contains the time correlations.	-	-
trans_id_in	long *	-	Transport format ID	-	Complete
time_ref_in	long *	-	Time reference ID	-	Any except XL_TIME_UNDEF
transport_in[dim]	long	See Table 3	Time in Transport format	See Table 3	See Table 3
trans_id_out	long *	-	Transport format ID	-	Complete
time_ref_out	long *	-	Time reference ID	-	Any except XL_TIME_UNDEF

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

- Transport format ID: trans_id_in and trans_id_out. Current document, section 6.2.
- Time reference ID: time_ref_in and time_ref_out. See [GEN_SUM].

Note that for the function to work correctly, the time references should be properly initialised before calling the function (see section 4.2. for details), unless time_ref_in = time_ref_out.

7.18.4Output Parameters

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

Table 59: Output parameters of xl_time_transport_to_transport

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_time_transport_to_transport	long	-	Status flag	-	-
transport_out[dim]	long	See Table 3	Time in Transport format	See Table 3	See Table 3
ierr	long	-	Error vector	-	-

7.18.5Warnings and Errors

Next table lists the possible error messages that can be returned by the **xl_time_transport_to_transport** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 error vector returned by the **xl_time_transport_to_transport** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM])

Table 60: Error messages of xl_time_transport_to_transport function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Input transport format ID is not correct	No calculation performed	XL_CFI_TIME_TRANS_TRANS_TRANS_IN_ERR	0
ERR	Input time reference ID is not correct	No calculation performed	XL_CFI_TIME_TRANS_TRANS_TIME_IN_ERR	1
ERR	Output transport format ID is not correct	No calculation performed	XL_CFI_TIME_TRANS_TRANS_TRANS_OUT_ERR	2
ERR	Output time reference ID is not correct	No calculation performed	XL_CFI_TIME_TRANS_TRANS_TIME_OUT_ERR	3
ERR	Number of days out of range	No calculation performed	XL_CFI_TIME_TRANS_TRANS_DAY_ERR	4
ERR	Number of seconds out of range	No calculation performed	XL_CFI_TIME_TRANS_TRANS_SEC_ERR	5
ERR	Number of milliseconds out of range	No calculation performed	XL_CFI_TIME_TRANS_TRANS_MILLISEC_ERR	6
ERR	Number of microseconds out of range	No calculation performed	XL_CFI_TIME_TRANS_TRANS_MICROSEC_ERR	7
ERR	Number of SIRAL extra counter ticks out of range	No calculation performed	XL_CFI_TIME_TRANS_TRANS_TICK_ERR	8
ERR	Time reference not initialised	No calculation performed	XL_CFI_TIME_TRANS_TRANS_REF_INIT_ERR	9
WARN	Time out of initialization range	Calculation performed. A message informs the user.	XL_CFI_TIME_TRANS_TRANS_REF_INIT_WARN	10
WARN	Bulletin A: previous computation perfromed inside file interval, current performed with formula	Calculation performed. A message informs the user.	XL_CFI_TIME_TRANS_TRANS_BUL_A_FORMULA_WARN	11
WARN	Bulletin B+A: current computation performed inside B-A gap. Previous computation was done inside B or A files intervals.	Calculation performed. A message informs the user.	XL_CFI_TIME_TRANS_TRANS_BUL_B_A_GAP_WARN	12
WARN	Previous computation performed inside initialization validity, current computation performed outside initialization validity	Calculation performed. A message informs the user.	XL_CFI_TIME_TRANS_TRANS_VALIDITY_WARN	13

7.19 xl_time_cuc_to_processing

7.19.1 Overview

The **xl_time_cuc_to_processing** CFI function transforms a time expressed in CCSDS UNSEGMENTED TIME CODE (CUC, see [CUC]) format into a time in Processing format.

7.19.2 CUC configuration

The input parameter of type xl_cuc_time_config tells the function how to make the transformation. The fields of this structure can take the following values:

- *cuc_type*: It is the type of CUC file used as input. It can take the values given by CUC time type enumeration, see section 6.2:
  - XL_CUC_T_FIELD: the input cuc_time contains only T-field octets.
  - XL_CUC_T_AND_P_FIELDS: the input cuc_time contains P-field and T-field octets (P-field octets before T-field octets).
- *epoch_type*: it is the epoch respect to which the CUC time is referenced. It can take the values given by CUC epoch type enumeration, see section 6.2:
  - XL_EPOCH_CCSDS: date 01/01/1958, 00h00
  - XL_EPOCH_GPS: date 6-Jan-1980, 00h00
  - XL_EPOCH_USER_DEFINED: defined by the user in *epoch* field (see below).

  This parameter is only relevant if *cuc_type* == XL_CUC_T_FIELD. Otherwise, the epoch type is taken from P field.
- *time_ref*: it is the time reference of the epoch type provided by user if *epoch_type* == XL_EPOCH_USER_DEFINED, or the epoch type read in P field is Level 2 Agency defined.
- *epoch*: it is the epoch type provided by user (in processing format) if *epoch_type* == XL_EPOCH_USER_DEFINED, or the epoch type read in P field is Level 2 Agency defined.
- *basic_time_unit_num_octets*: it is the number of unit octets in input cuc_time. Only relevant if *cuc_type* == XL_CUC_T_FIELD. Otherwise the number is taken from P field.
- *fractional_time_unit_num_octets*: it is the number of fraction of unit octets in input cuc_time. Only relevant if *cuc_type* == XL_CUC_T_FIELD. Otherwise the number is taken from P field.

7.19.3 Calling Interface

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

```
#include <explorer_lib.h>
{
      double processing_out;
      xl_time_id time_id = {NULL};
```

```
        long ierr[XL_NUM_ERR_CUC_PROC], status;
        xl_cuc_time_config config;
        unsigned char cuc_time[XL_MAX_CUC_ARRAY_LENGTH];
        long time_ref;


        status =    xl_time_cuc_to_processing(&time_id, &config,
                                              cuc_time, time_ref,
                                              &processing_out, ierr);
}
```

The `XL_MAX_CUC_ARRAY_LENGTH` and `XL_NUM_ERR_CUC_PROC` constants are defined in the file *explorer_lib.h.*

7.19.4 Input Parameters

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

Table 61: Input parameters of xl_time_cuc_to_processing function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
time_id	xl_time_id*	-	Structure that contains the time correlations.	-	-
config	xl_cuc_time_config	-	CUC time configuration	-	-
cuc_time	unsigned char*	-	CUC time	-	-
time_ref	long	-	Time reference ID	-	Complete

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

- Time reference ID: time_ref. See [GEN_SUM].

Note that for the function to work correctly, the time references should be properly initialised before calling the function (see section 4.2. for details).

7.19.5 Output Parameters

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

Table 62: Output parameters of xl_time_cuc_to_processing

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_time_cuc_to_processing	long	-	Status flag	-	-
processing_out	double*	-	Time in Processing Format	Decimal days, MJD2000 (Processing)	[-18262.0,36524.0]

ierr	long	-	Error vector	-	-

7.19.6Warnings and Errors

Next table lists the possible error messages that can be returned by the **xl_time_cuc_to_processing** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 by translating the error vector returned by the **xl_time_cuc_to_processing** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM])

Table 63: Error messages of xl_time_cuc_to_processing function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Error in CUC configuration	No calculation performed	XL_CFI_TIME_CUC_TO_PROCESSING_WRONG_CONFIG_ERR	0
ERR	Error getting CUC P field data	No calculation performed	XL_CFI_TIME_CUC_TO_PROCESSING_GET_P_FIELD_DATA_ERR	1
ERR	Error getting CUC epoch	No calculation performed	XL_CFI_TIME_CUC_TO_PROCESSING_GET_CUC_EPOCH_ERR	2

7.20 xl_time_processing_to_cuc

7.20.1 Overview

The **xl_time_processing_to_cuc** CFI function transforms a time expressed processing format to CCSDS UNSEGMENTED TIME CODE (CUC, see [CUC]).

7.20.2 CUC configuration

The input parameter of type *xl_cuc_time_config* tells the function how to make the transformation. The fields of this structure can take the following values:

- *cuc_type*: It is the type of CUC file used as input. It can take the values given by CUC time type enumeration, see section 6.2:
  - XL_CUC_T_FIELD: the output cuc_time will contain only T-field octets.
  - XL_CUC_T_AND_P_FIELDS: the output cuc_time will contains P-field and T-field octets (P-field octets before T-field octets).
- *epoch_type*: it is the epoch respect to which the CUC time is referenced. It can take the values given by CUC epoch type enumeration, see section 6.2:
  - XL_EPOCH_CCSDS: date 01/01/1958, 00h00

- XL_EPOCH_GPS: date 6-Jan-1980, 00h00
- XL_EPOCH_USER_DEFINED: defined by the user in *epoch* field (see below).

- *time_ref*: it is the time reference of the epoch type provided by user if *epoch_type* == XL_EPOCH_USER_DEFINED.
- *epoch*: it is the epoch type provided by user (in processing format) if *epoch_type* == XL_EPOCH_USER_DEFINED.
- *basic_time_unit_num_octets*: it is the number of unit octets in output cuc_time.
- *fractional_time_unit_num_octets*: it is the number of fraction of unit octets in output cuc_time.
- Note: P field octets (one or two) are automatically computed and added by the function, depending on the previous information.

7.20.3 Calling Interface

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

```
#include <explorer_lib.h>
{
      double processing_in;
      xl_time_id time_id = {NULL};
      long ierr[XL_NUM_ERR_PROC_CUC], status;
      xl_cuc_time_config config;
      unsigned char cuc_time[XL_MAX_CUC_ARRAY_LENGTH];
      long time_ref;

      status =    xl_time_cuc_to_processing(&time_id, &config,
                                            time_ref,processing_in,
                                            cuc_time, ierr);
}
```

The `XL_MAX_CUC_ARRAY_LENGTH` and `XL_NUM_ERR_PROC_CUC` constants are defined in the file *explorer_lib.h.*

7.20.4 Input Parameters

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

Table 64: Input parameters of xl_time_processing_to_cuc function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
time_id	xl_time_id*	-	Structure that contains the time	-	-

			correlations.		
config	xl_cuc_time_config	-	CUC time configuration	-	-
time_ref	long	-	Time reference ID	-	Complete
processing_in	double*	-	Time in Processing Format	Decimal days, MJD2000 (Processing)	[-18262.0,36524.0]

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

- Time reference ID: time_ref. See [GEN_SUM].

Note that for the function to work correctly, the time references should be properly initialised before calling the function (see section 4.2. for details).

7.20.5 Output Parameters

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

Table 65: Output parameters of xl_time_processing_to_cuc

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_time_processing_to_cuc	long	-	Status flag	-	-
cuc_time	unsigned char*	-	CUC time	-	-
ierr	long	-	Error vector	-	-

7.20.6 Warnings and Errors

Next table lists the possible error messages that can be returned by the **xl_time_processing_to_cuc** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 by translating the error vector returned by the **xl_time_processing_to_cuc** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM])

Table 66: Error messages of xl_time_processing_to_cuc function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Error in CUC configuration	No calculation performed	XL_CFI_TIME_PROCESSING_TO_CUC_WRONG_CONFIG_ERR	0
ERR	Error getting CUC epoch	No calculation performed	XL_CFI_TIME_PROCESSING_TO_CUC_GET_CUC_EPOCH_ERR	1
ERR	CUC epoch must be previous	No calculation performed	XL_CFI_TIME_PROCESSIN	2

	to processing input date		G_TO_CUC_WRONG_EPOCH_ERR	
ERR	Error computing P-field	No calculation performed	XL_CFI_TIME_PROCESSING_TO_CUC_COMPUTE_P_FIELD_ERR	

7.21 xl_time_add

7.21.1Overview

The **xl_time_add** CFI function adds a time duration to a TAI, UTC, UT1 or GPS times expressed in Processing format.

User should be aware that the use of UTC in Processing format is not encouraged, due to the discontinuty that is caused by the introduction of leap seconds. See [IERS] for further details.

7.21.2Calling interface

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

```
#include <explorer_lib.h>
{
      long proc_id, time_ref;
      double processing_in, added_duration, processing_out;
      long ierr[XL_NUM_ERR_TIME_ADD], status;

      status =    xl_time_add      (&proc_id, &time_ref,
                        &processing_in, &added_duration,
                        &processing_out, ierr);
}
```

The `XL_NUM_ERR_TIME_ADD` constant is defined in the file *explorer_lib.h.*

7.21.3Input parameters

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

Table 67: Input parameters of xl_time_add function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
proc_id	long *	-	Processing format ID	-	Complete
time_ref	long *	-	Time reference ID	-	Any except

					XL_TIME_UNDEF
processing_in	double*	-	Time in Processing Format	Decimal days, MJD2000 (Processing format)	[-18262.0,36524.0]
added_duration	double*	-	Duration to be added	Decimal days (Processing format)	-

It is important to point out that the duration is not a time, but a time interval expressed in decimal days to be added to the original time.

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

- Processing format ID: proc_id. Current document, section 6.2.
- Time reference ID: time_ref. See [GEN_SUM].

7.21.4Output parameters

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

Table 68: Output parameters of xl_time_add function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_time_add	long	-	Status flag	-	-
processing_out	double*	-	Time in Processing Format	Decimal days (Processing format)	[-18262.0,36524.0]
ierr	long	-	Error vector	-	-

7.21.5Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_time_add** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 by translating the error vector returned by the **xl_time_add** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 69: Error messages of xl_time_add function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Processing format ID is not correct	No calculation performed	XL_CFI_TIME_ADD_PROC_ERR	0
ERR	Time reference ID is not correct	No calculation performed	XL_CFI_TIME_ADD_TIME_ERR	1
ERR	Input processing time is out of	No calculation performed	XL_CFI_TIME_ADD_DAY_I	2

	range		N_ERR	
ERR	Output processing time is out of range	No calculation performed	XL_CFI_TIME_ADD_DAY_OUT_ERR	3

7.22xl_time_diff

7.22.1Overview

The **xl_time_diff** CFI function calculates the time difference between two TAI, UTC, UT1 or GPS times expressed in Processing format.

User should be aware that the use of UTC in Processing format is not encouraged, due to the discontinuity that is caused by the introduction of leap seconds. See [IERS] for further details.

7.22.2Calling interface

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

```
#include <explorer_lib.h>
{
      long proc_id, time_ref;
      double processing_in_1, processing_in_2, processing_out;
      long ierr[XL_NUM_ERR_TIME_DIFF], status;

      status =    xl_time_diff    (&proc_id, &time_ref,
                        &processing_in_1, &processing_in_2,
                        &processing_out, ierr);
}
```

Note that `processing_out` is a duration, not a time itself, so it should not be converted to another reference or format.

The `XL_NUM_ERR_TIME_DIFF` constant is defined in the file *explorer_lib.h*.

7.22.3Input parameters

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

Table 70: Input parameters of xl_time_diff function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
proc_id	long *	-	Processing format ID	-	Complete
time_ref	long *	-	Time reference ID	-	Any except XL_TIME_UNDEF
processing_in_1	double*	-	Time in Processing Format	Decimal days, MJD2000 (Processing format)	[-18262.0,36524.0]

processing_in_2	double*	-	Time in Processing Format	Decimal days, MJD2000 (Processing format)	[-18262.0,36524.0]

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

- Processing format ID: proc_id. Current document, section 6.2.
- Time reference ID: time_ref. See [GEN_SUM].

7.22.4Output parameters

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

Table 71: Output parameters of xl_time_diff function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_time_diff	long	-	Status flag	-	-
processing_out	double*	-	Time difference between processing_in_1 and processing_in_2 expressed in decimal days	Decimal days (Processing format)	-
ierr	long	-	Error vector	-	-

7.22.5Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_time_diff** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 by translating the error vector returned by the **xl_time_diff** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 72: Error messages of xl_time_diff function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Processing format ID is not correct	No calculation performed	XL_CFI_TIME_DIFF_PROC_ERR	0
ERR	Time reference ID is not correct	No calculation performed	XL_CFI_TIME_DIFF_TIME_ERR	1
ERR	Input processing time #1 is out of range	No calculation performed	XL_CFI_TIME_DIFF_DAY_IN_1_ERR	2
ERR	Input processing time #2 is out of range	No calculation performed	XL_CFI_TIME_DIFF_DAY_IN_2_ERR	3

7.23xl_time_obt_to_time

7.23.1Overview

The **xl_time_obt_to_time** CFI function transforms from On-board Time (OBT) count to UTC Processing time.

User should be aware that the use of UTC in Processing format is not encouraged, due to the discontinuity that is caused by the introduction of leap seconds. See [IERS] for further details.

See [MCD] or details on time formats and representations, in particular the definition of OBT.

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

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

The **xl_time_obt_to_time** CFI function applies to satellites where OBT time is a counter, which needs to be correlated to an actual time reference. Nevertheless, some other satellites, like Cryosat, use an actual time reference on-board. In this case, the on-board time conversions are handled by the **xl_time_processing_to_processing** function.

Due to the different OBT models used by the various spacecraft, specific data structures are used for each of them. The keep a single interface for the function, a void pointer is used to pass the specific structures to the generic function.

The following data structures are defined for ENVISAT:

```
/* Envisat OBT Structure */
typedef struct
{
 long          sat_id;
 double        time0;
 unsigned long obt0[2];
 unsigned long period0;
} xl_envisat_obt_param;

typedef struct
{
 long          sat_id;
 unsigned long obt[2];
} xl_envisat_obt_value;
```

for GOCE:

```
/* GOCE OBT Structure */
typedef struct
{
```

```
 long          sat_id;
 unsigned long utc0_c;
 unsigned int  utc0_f;
 unsigned long obt0_c;
 unsigned int  obt0_f;
 double        gradient;
 double        offset;
} xl_goce_obt_param;


typedef struct
{
 long          sat_id;
 double        obt;
} xl_goce_obt_value;
```

for SMOS

```
typedef struct
{
 long sat_id;
 long delta_seconds; /* number of seconds to be applied to UTC to
                        give UTC Proteus (just in case UTC Proteus
                        reference is actually GPS Time)*/
 unsigned long obet0_c; /* OBET Coarse Time (in seconds) */
 unsigned long obet0_f; /* OBET Fine Time */
 unsigned long utc0_week; /* UTC (Proteus format) week number */
 unsigned long utc0_seconds; /* UTC (Proteus format) seconds of
                                week */
 unsigned long utc0_fraction; /* UTC (Proteus format) fraction of
                                 seconds */
} xl_smos_obt_param;


typedef struct
{
 long sat_id;
 unsigned long obet_c; /* OBET Coarse Time (in seconds) */
 unsigned long obet_f; /* OBET Fine Time */
} xl_smos_obt_value;
```

and for ADM

```
typedef struct
{
 long sat_id;
 long delta_seconds; /* it refers to the number of seconds to be
                        applied to UTC to give GPS (GPST - UTC) */
} xl_adm_obt_param;


typedef struct
{
 long sat_id;
 unsigned long cuc_sec; /* CCSDS Unsegmented Time Code (secs) */
 unsigned long cuc_subsec; /* CCSDS Unsegmented Time Code
                              (subseconds) */
} xl_adm_obt_value;
```

The sat_id parameter within the stucture has to be assigned equal to the sat_id passed to the function.

7.23.2Calling interface

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

```
#include <explorer_lib.h>
{
      long sat_id, proc_id;
      xl_envisat_obt_param obt_param; /*example for ENVISAT */
      xl_envisat_obt_value obt_value_in; /*example for ENVISAT */
      double time_out;
      long ierr[XL_NUM_ERR_OBT_TIME], status;


      status =    xl_time_obt_to_time (&sat_id,
                                       &proc_id,
                                       &obt_param,
                                       &obt_value_in,
                                       &time_out,
                                       ierr);
```

```
        /* Or, using the run_id */
        long run_id;
        status =     xl_time_obt_to_time_run (&run_id,
                                              &proc_id,
                                              &obt_param,
                                              &obt_value_in,
                                              6time_out,
                                              err);
}
```

The XL_NUM_ERR_OBT_TIME constant is defined in the file *explorer_lib.h.*

7.23.3Input parameters

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

Table 73: Input parameters of xl_time_obt_to_time function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
sat_id	long *	-	Satellite ID	-	Complete
proc_id	long *	-	Processing format ID	-	Complete
obt_param	void *	-	Pointer to xl_<satellite>_obt_param	-	-
obt_value_in	void *	-	Pointer to xl_<satellite>_obt_value	-	-

Table 74: Input parameters of xl_envisat_obt_param structure

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
sat_id	long	-	Satellite ID	-	XL_SAT_ENVISAT
time0	double	-	Reference time	Decimal days (Processing format)	[-18262.0,36524.0]
obt0	unsigned long[2]	-	Array of counters containing the OBT at the reference time (in the satellite dependant format)	TBD	TBD
period0	unsigned long	-	Actual on-board clock period	TBD	TBD

Table 75: Input parameters of xl_envisat_obt_value structure

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
sat_id	long	-	Satellite ID	-	XL_SAT_ENVISAT
obt	unsigned long[2]	-	Array of counters containing the OBT time (in the satellite dependant format)	TBD	TBD

Table 76: Input parameters of xl_goce_obt_param structure

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
sat_id	long	-	Satellite ID	-	XL_SAT_GOCE
utc0_c	unsigned long	-	Coarse UTC0	seconds	>=0
utc0_f	unsigned int	-	Fine UTC0	2^{-16} seconds	>=0
obt0_c	unsigned long	-	Coarse OBT0	seconds	>=0
obt0_f	unsigned int	-	Fine OBT0	2^{-16} seconds	>=0
gradient	double	-	Gradient between the OBT and the UTC	-	-
offset	double	-	Offset between the OBT and the UTC	seconds	-

Table 77: Input parameters of xl_goce_obt_value structure

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
sat_id	long	-	Satellite ID	-	XL_SAT_GOCE
obt	double	-	OBT time	sconds	-

Table 78: Input parameters of xl_smos_obt_param structure

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
sat_id	long	-	Satellite ID	-	XL_SAT_SMOS
delta_seconds	long	-	Number of seconds to be applied to UTC to give UTC Proteus (in case UTC Proteus reference is actually GPS Time)	seconds	
obet0_c	unsigned long	-	OBET0 Coarse Time	seconds	>=0
obet0_f	unsigned long	-	OBET 0Fine Time	2^{-16} seconds	>=0
utc0_week	unsigned long	-	UTC0 (Proteus format) week number	weeks	>=0

utc0_second	unsigned long	-	UTC0 (Proteus format) seconds of week	seconds	>=0
utc0_fraction	unsigned long	-	UTC0 (Proteus format) fraction of seconds	2^{-16} seconds	>=0

Table 79: Input parameters of xl_smos_obt_value structure

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
sat_id	long	-	Satellite ID	-	XL_SAT_SMOS
obet_c	unsigned long	-	OBET Coarse Time	seconds	>=0
obet_f	unsigned long	-	OBET Fine Time	2^{-16} seconds	>=0

Table 80: Input parameters of xl_adm_obt_param structure

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
sat_id	long	-	Satellite ID	-	XL_SAT_ADM
delta_seconds	long	-	Number of seconds to be applied to UTC to give GPS (GPST - UTC)	seconds	

Table 81: Input parameters of xl_adm_obt_value structure

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
sat_id	long	-	Satellite ID	-	XL_SAT_ADM
cuc_sec	unsigned long	-	CCSDS Unsegmented Time Code (seconds)	seconds	>=0
cuc_subsec	unsigned long	-	CCSDS Unsegmented Time Code (subseconds)	2^{-16} seconds	>=0

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

- Satellite ID: sat_id. See [GEN_SUM].
- Processing format ID: proc_id. Current document, section 6.2.

7.23.4Output parameters

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

Table 82: Output parameters of xl_time_obt_to_time function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_time_obt_to_time	long	-	Status flag	-	-
time_out	double*	-	UTC Time in Processing Format	Decimal days, MJD2000 (Processing format)	[-18262.0,36524.0]
ierr	long	-	Error vector	-	-

7.23.5Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_time_obt_to_time** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 by translating the error vector returned by the **xl_time_obt_to_time** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 83: Error messages of xl_time_obt_to_time function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Satellite ID is not correct	No calculation performed	XL_CFI_TIME_OBT_TIME_SAT_ERR	0
ERR	Processing format ID is not correct	No calculation performed	XL_CFI_TIME_OBT_TIME_PROC_ERR	1
ERR	Structure inconsistent with Satellite ID	No calculation performed	XL_CFI_TIME_OBT_TIME_INCONSISTENT_STRUCT_ERR	2
ERR	Input reference time is out of range	No calculation performed	XL_CFI_TIME_OBT_TIME_DAY_REF_ERR	3
ERR	No OBT defined for this satellite ID	No calculation performed	XL_CFI_TIME_OBT_TIME_OBT_SAT_ERR	4
ERR	OBT at reference time is out of allowed range	No calculation performed	XL_CFI_TIME_OBT_TIME_OBT_ERR	5
ERR	Period of the On-Board clock is null	No calculation performed	XL_CFI_TIME_OBT_TIME_CLOCK_ERR	6
ERR	Output time is out of range	No calculation performed	XL_CFI_TIME_OBT_TIME_DAY_OUT_ERR	7

7.24xl_time_time_to_obt

7.24.1Overview

The **xl_time_time_to_obt** CFI function transforms a UTC Processing time to OBT count.

User should be aware that the use of UTC in Processing format is not encouraged, due to the discontinuty that is caused by the introduction of leap seconds. See [IERS] for further details.

See [MCD] for details on time formats and representations, in particular the definition OBT.

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

The **xl_time_time_to_obt** CFI function applies to satellites where OBT time is a counter, which needs to be correlated to an actual time reference. Nevertheless, some other satellites, like Cryosat, use an actual time reference on-board. In this case, the on-board time conversions are handled by the **xl_time_processing_to_processing** function.

Due to the different OBT models used by the various spacecraft, specific data structures are used for each of them. The keep a single interface for the function, a void pointer is used to pass the specific structures to the generic function.

The following data structures are defined for ENVISAT:

```
/* Envisat OBT Structure */
typedef struct
{
 long          sat_id;
 double        time0;
 unsigned long obt0[2];
 unsigned long period0;
} xl_envisat_obt_param;

typedef struct
{
 long          sat_id;
 unsigned long obt[2];
} xl_envisat_obt_value;
```

for GOCE:

```
/* GOCE OBT Structure */
typedef struct
{
 long          sat_id;
 unsigned long utc0_c;
 unsigned int  utc0_f;
```

```
 unsigned long obt0_c;
 unsigned int  obt0_f;
 double        gradient;
 double        offset;
} xl_goce_obt_param;


typedef struct
{
 long          sat_id;
 double        obt;
} xl_goce_obt_value;
```

for SMOS

```
typedef struct
{
 long sat_id;
 long delta_seconds; /* number of seconds to be applied to UTC to
                        give UTC Proteus (just in case UTC Proteus
                        reference is actually GPS Time)*/
 unsigned long obet0_c; /* OBET Coarse Time (in seconds) */
 unsigned long obet0_f; /* OBET Fine Time */
 unsigned long utc0_week; /* UTC (Proteus format) week number */
 unsigned long utc0_seconds; /* UTC (Proteus format) seconds of
                                week */
 unsigned long utc0_fraction; /* UTC (Proteus format) fraction of
                                 seconds */
} xl_smos_obt_param;


typedef struct
{
 long sat_id;
 unsigned long obet_c; /* OBET Coarse Time (in seconds) */
 unsigned long obet_f; /* OBET Fine Time */
} xl_smos_obt_value;
```

and for ADM

```
typedef struct
```

```
{
 long sat_id;
 long delta_seconds; /* it refers to the number of seconds to be
                        applied to UTC to give GPS (GPST - UTC) */
} xl_adm_obt_param;


typedef struct
{
 long sat_id;
 unsigned long cuc_sec; /* CCSDS Unsegmented Time Code (secs) */
 unsigned long cuc_subsec; /* CCSDS Unsegmented Time Code
                              (subseconds) */
} xl_adm_obt_value;
```

The sat_id parameter within the stucture has to be assigned equal to the sat_id passed to the function.

7.24.2Calling interface

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

```
#include <explorer_lib.h>
{
      long sat_id, proc_id;
      double time_in;
      xl_envisat_obt_param obt_param; /*example for ENVISAT */
      xl_envisat_obt_value obt_value_out; /*example for ENVISAT */
      long ierr[XL_NUM_ERR_TIME_OBT], status;

      status =    xl_time_time_to_obt (&sat_id,
                                       &proc_id,
                                       &obt_param,
                                       &time_in,
                                       &obt_value_out,
                                       ierr);
```

```
        /* Or, using the run_id */
        long run_id;


        status =     xl_time_time_to_obt_run (&run_id,
                                              &proc_id,
                                              &obt_param,
                                              &time_in,
                                              &obt_value_out,
                                              ierr);
}
```

The `XL_NUM_ERR_TIME_OBT` constant is defined in the file *explorer_lib.h.*

7.24.3Input parameters

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

Table 84: Input parameters of xl_time_obt_to_time function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
sat_id	long *	-	Satellite ID	-	Complete
proc_id	long *	-	Processing format ID	-	Complete
obt_param	void *	-	Pointer to xl_<satellite>_obt_param	-	-
time_in	double*	-	UTC Time	Decimal days (Processing format)	[-18262.0,36524.0]

Table 85: Input parameters of xl_envisat_obt_param structure

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
sat_id	long	-	Satellite ID	-	XL_SAT_ENVISAT
time0	double	-	Reference time	Decimal days (Processing format)	[-18262.0,36524.0]
obt0	unsigned long[2]	-	Array of counters containing the OBT at the reference time (in the satellite dependant format)	TBD	TBD
period0	unsigned long	-	Actual on-board clock period	TBD	TBD

Table 86: Input parameters of xl_goce_obt_param structure

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
sat_id	long	-	Satellite ID	-	XL_SAT_GOCE
utc0_c	unsigned long	-	Coarse UTC0	seconds	>=0
utc0_f	unsigned int	-	Fine UTC0	2^{-16} seconds	>=0
obt0_c	unsigned long	-	Coarse OBT0	seconds	>=0
obt0_f	unsigned int	-	Fine OBT0	2^{-16} seconds	>=0
gradient	double	-	Gradient between the OBT and the UTC	-	-
offset	double	-	Offset between the OBT and the UTC	seconds	-

Table 87: Input parameters of xl_smos_obt_param structure

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
sat_id	long	-	Satellite ID	-	XL_SAT_SMOS
delta_seconds	long	-	Number of seconds to be applied to UTC to give UTC Proteus (in case UTC Proteus reference is actually GPS Time)	seconds	
obet0_c	unsigned long	-	OBET0 Coarse Time	seconds	>=0
obet0_f	unsigned long	-	OBET 0Fine Time	2^{-16} seconds	>=0
utc0_week	unsigned long	-	UTC0 (Proteus format) week number	weeks	>=0
utc0_second	unsigned long	-	UTC0 (Proteus format) seconds of week	seconds	>=0
utc0_fraction	unsigned long	-	UTC0 (Proteus format) fraction of seconds	2^{-16} seconds	>=0

Table 88: Input parameters of xl_adm_obt_param structure

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
sat_id	long	-	Satellite ID	-	XL_SAT_ADM
delta_seconds	long	-	Number of seconds to be applied to UTC to give GPS (GPST - UTC)	seconds	

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

- Satellite ID: sat_id. See [GEN_SUM].
- Time reference ID: time_ref. See [GEN_SUM].

7.24.4Output parameters

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

Table 89: Output parameters of xl_time_time_to_obt function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_time_time_to_obt	long	-	Status flag	-	-
obt_value_out	void *	-	Pointer to xl_<satellite>_obt_value	-	-
ierr	long	-	Error vector	-	-

Table 90: Output parameters of xl_envisat_obt_value structure

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
sat_id	long	-	Satellite ID	-	XL_SAT_ENVISAT
obt	unsigned long[2]	-	Array of counters containing the OBT time (in the satellite dependant format)	TBD	TBD

Table 91: Output parameters of xl_goce_obt_value structure

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
sat_id	long	-	Satellite ID	-	XL_SAT_GOCE
obt	double	-	OBT time	sconds	-

Table 92: Output parameters of xl_smos_obt_value structure

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
sat_id	long	-	Satellite ID	-	XL_SAT_SMOS
obet_c	unsigned long	-	OBET Coarse Time	seconds	>=0
obet_f	unsigned long	-	OBET Fine Time	2^{-16} seconds	>=0

Table 93: Output parameters of xl_adm_obt_value structure

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
sat_id	long	-	Satellite ID	-	XL_SAT_ADM
cuc_sec	unsigned long	-	CCSDS Unsegmented Time Code (seconds)	seconds	>=0
cuc_subsec	unsigned long	-	CCSDS Unsegmented Time Code (subseconds)	2^{-16} seconds	>=0

7.24.5Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_time_time_to_obt** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 by translating the error vector returned by the **xl_time_time_to_obt** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 94: Error messages of xl_time_time_to_obt function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Satellite ID is not correct	No calculation performed	XL_CFI_TIME_TIME_OBT_SAT_ERR	0
ERR	Processing format ID is not correct	No calculation performed	XL_CFI_TIME_TIME_OBT_PROC_ERR	1
ERR	Structure inconsistent with Satellite ID	No calculation performed	XL_CFI_TIME_TIME_OBT_INCONSISTENT_STRUCT_ERR	2
ERR	Input time is out of range	No calculation performed	XL_CFI_TIME_TIME_OBT_DAY_IN_ERR	3
ERR	Input reference time is out of range	No calculation performed	XL_CFI_TIME_TIME_OBT_DAY_REF_ERR	4
ERR	No OBT defined for this satellite ID	No calculation performed	XL_CFI_TIME_TIME_OBT_OBT_SAT_ERR	5
ERR	OBT at reference time is out of allowed range	No calculation performed	XL_CFI_TIME_TIME_OBT_OBT_ERR	6
ERR	Period of the On-Board clock is null	No calculation performed	XL_CFI_TIME_TIME_OBT_CLOCK_ERR	7

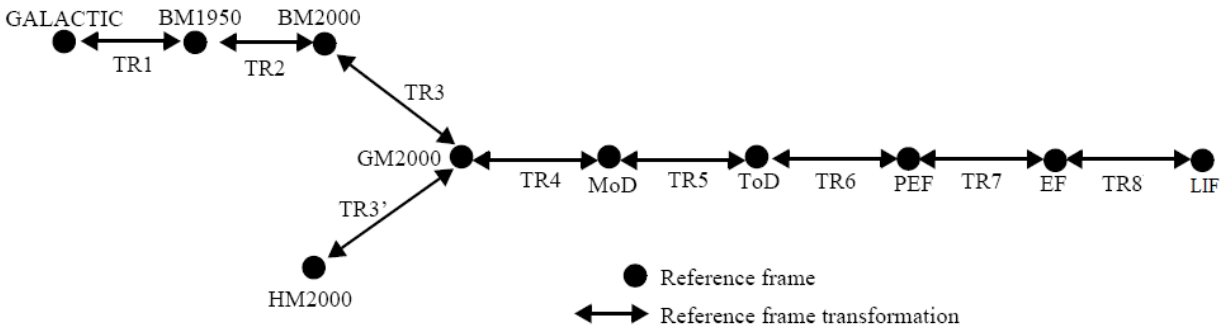
7.25xl_change_cart_cs

7.25.1Overview

The **xl_change_cart_cs** CFI function transforms a cartesian state vector between different reference frames. The transformation are done sequentially following the schema in figure . Note that the transformations between BM2000, HM2000 and GM2000 involve a translation of the input vectors (TR3 and TR3'). In case that the input vector is a direction and not a location, the transformation should only apply the rotations between frames, so the transformation should be done in several steps with **xl_change_cart_cs**, skipping the translations[1].

Note: conversion from/to LIF frame is not enabled by default. To enable it, the user needs to set the enable flag and the reference longitude and UTC time (see [MCD]). This can be done as follows:

- read time_id data using the xl_time_get_id_data function;
- fill the structure launch_inertial_frame_config within the time_id data;
- set the modified time_id data using function xl_time_set_id_data.



Reference frames:

GALACTIC = Galactic CS (see section 5.1.1)
BM1950 = Barycentric Mean of 1950.0 (see section 5.1.2)
BM2000 = Barycentric Mean of 2000.0 (see section 5.1.3)
HM2000 = Heliocentric Mean of 2000.0 (see section 5.1.4)
GM2000 = Geocentric Mean of 2000.0 (see section 5.1.5)
MoD = Mean of Date (see section 5.1.6)
ToD = True of Date (see section 5.1.7)
PEF = Pseudo Earth Fixed (see section 5.1.8)
EF = Earth Fixed (see section 5.1.9)
LIF = Launch Inertial Frame (see section 5.1.11)

Transfromations:

TR1 = Galactic to Barycentric Mean of 1950 (see section 5.3.1)
TR2 = Barycentric 1950 to Barycentric 2000 (see section 5.3.2)
TR3 = Solar system barycentre to Earth centre translation (see section 5.3.3)
TR3' = Sun centre to Earth centre translation (see section 5.3.4)
TR4 = Precession (see section 5.3.5)
TR5 = Nutation (see section 5.3.6)
TR6 = Earth rotation + nutation term (see section 5.3.7)
TR7 = Polar motion rotation (see section 5.3.8)
TR8 = Earth rotation

Figure 2: Change cartesian coordinates

1 For this purposoe it is also possible to use the CFI function **xp_change_frame** in the eo_pointing library

7.25.2Calling interface

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

```
#include <explorer_lib.h>
{
      long mode, cs_in, cs_out, time_ref;
      double time;
      double pos[3], vel[3], acc[3];
      double pos_out[3], vel_out[3], acc_out[3];
      xl_model_id model_id = {NULL};
      xl_time_id time_id = {NULL};
      long status;

      status =    xl_change_cart_cs (&model_id, &time_id,
                                     &mode, &cs_in, &cs_out,
                                     &time_ref, &time, pos, vel, acc,
                                     pos_out,vel_out, acc_out);

      /* Or, using the run_id */
      long run_id;

      status =    xl_change_cart_cs_run (&run_id, &mode, &cs_in,
&cs_out,
                                         &time_ref, &time, pos, vel, acc,
                                         pos_out,vel_out, acc_out);
}
```

7.25.3Input parameters

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

Table 95: Input parameters of xl_change_cart_cs function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
model_id	xl_model_id*	-	Model ID	-	-
time_id	xl_time_id*	-	Structure that contains the time correlations.	-	-

mode	long*	-	Calculation mode selection		Complete
cs_in	long *	-	Initial reference frame ID	-	Complete
cs_out	long *	-	Final reference frame ID	-	Complete
time_ref	long *	-	Time reference ID	-	Any except XL_TIME_UNDEF
time	double*	-	Reference time	Decimal days (Processing format)	[-18262.0,36524.0]
pos[3]	double	all	Input position vector (Initial reference frame)	m	-
vel[3]	double	all	Input velocity vector (Initial reference frame) This value is dummy if *mode* is·XL_CALC_POS except for the transformations between BM1950 and BM2000	m/s	-
acc[3]	double	all	Input acceleration vector (Initial reference frame) Dummy if *mode* is either: · XL_CALC_POS · XL_CALC_POS_VEL	m/s^2	-

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

- Calculation mode selection: mode. See current document, section 6.2.
- Time reference ID: time_ref. See [GEN_SUM].
- Reference frame: cs_in, cs_out. See current document, section 6.2.

Notes:

- the function could not work correctly if the time references are not properly initialised before calling the function (see section 4.2. for details).
- For objects located closer to 1 AU, the transformation from and to BM1950 may produce incorrect results

7.25.4Output parameters

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

Table 96: Output parameters of xl_change_cart_cs function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_change_cart_cs	long	-	Extended status flag	-	-
pos_out[3]	double	all	Output position vector (Final reference frame)	m	-
vel_out[3]	double	all	Output velocity vector (Final reference frame) Returned only if *mode* is either: · XL_CALC_POS_VEL	m/s	-

			· XL_CALC_POS_VEL_ACC		
acc_out[3]	double	all	Output acceleration vector (Final reference frame) Returned only if *mode* is: · XL_CALC_POS_VEL_ACC	m/s^2	-

7.25.5Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_change_cart_cs** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 **xl_change_cart_cs** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 97: Error messages of xl_change_cart_cs function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Input time reference ID is not correct	No calculation performed	XL_CFI_CHANGE_CART_CS_REF_ERR	0
ERR	Input date is out of range	No calculation performed	XL_CFI_CHANGE_CART_CS_DAY_ERR	1
ERR	Calculation mode ID is not correct	No calculation performed	XL_CFI_CHANGE_CART_CS_MODE_ERR	2
ERR	Input reference frame is not correct	No calculation performed	XL_CFI_CHANGE_CART_CS_INPUT_CS_ERR	3
ERR	Output reference frame is not correct	No calculation performed	XL_CFI_CHANGE_CART_CS_OUTPUT_CS_ERR	4
ERR	Time Reference not initialised	No calculation performed	XL_CFI_CHANGE_CART_CS_REF_INIT_ERR	5
WARN	Bulletin A: previous computation perfromed inside file interval, current performed with formula	Calculation performed	XL_CHANGE_CART_CS_BUL_A_FORMULA_WARN	6
WARN	Bulletin B+A: current computation performed inside B-A gap. Previous computation was done inside B or A files intervals.	Calculation performed	XL_CHANGE_CART_CS_BUL_B_A_GAP_WARN	7
WARN	Previous computation performed inside initialization validity, current computation performed outside initialization validity	Calculation performed	XL_CHANGE_CART_CS_VALIDITY_WARN	8
ERR	Error computing TAI time.	No calculation performed	XL_CHANGE_CART_CS_TIME_COMPUTATION_ERR	9
ERR	Error computing transformation.	No calculation performed	XL_CHANGE_CART_CS_CHANGE_CS_ERR	10
ERR	Conversion to/from LIF requires time id initialised with EOGS.	No calculation performed	XL_CHANGE_CART_CS_EOGS_NOT_INITIALISED_ERR	11

7.26xl_geod_to_cart

7.26.1Overview

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

7.26.2Calling interface

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

```
#include <explorer_lib.h>
{
        long mode;
        xl_model_id model_id = {NULL};
        double lon, lat, h, lon_rate, lat_rate, h_rate;
        double pos[3], vel[3];
        long status;

        status =     xl_geod_to_cart (&model_id, &mode, &lon, &lat, &h,
                                      &lon_rate, &lat_rate, &h_rate,
                                      pos, vel);

}
```

7.26.3Input parameters

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

Table 98: Input parameters of xl_geod_to_cart function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
model_id	xl_model_id	-	Model ID	-	-
mode	long*	-	Calculation mode selection	-	Select either: · XL_CALC_POS · XL_CALC_POS_VEL
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 >= $-b_{ellipsoid}$ (sat_id dependent)
lon_rate	double *	-	Geocentric longitude rate (Earth fixed CS)	deg/s	-

			Dummy if *mode* is: · XL_CALC_POS		
lat_rate	double *	-	Geodetic latitude rate (Earth fixed CS) Dummy if *mode* is: · XL_CALC_POS	deg/s	-
h_rate	double *		Geodetic altitude rate (Earth fixed CS) Dummy if *mode* is: · XL_CALC_POS	m/s	-

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

- Calculation mode selection: mode. See current document, section 6.2.

7.26.4Output parameters

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

Table 99: Output parameters of xl_geod_to_cart function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_geod_to_cart	long	-	Extended status flag	-	-
pos[3]	double	all	Cartesian position vector (Earth fixed CS)	m	-
vel[3]	double	all	Cartesian velocity vector (Earth fixed CS) Returned only if *mode* is: · XL_CALC_POS_VEL	m/s	-

7.26.5Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_geod_to_cart** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 **xl_geod_to_cart** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 100: Error messages of xl_geod_to_cart function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Wrong geodetic latitude on input (out of range)	No calculation performed	XL_CFI_GEOD_CART_EL_GT_90_ERR	0
WARN	Calculation mode ID is not correct	Calculation performed. A message informs the user.	XL_CFI_GEOD_CART_MODE_WARN	1

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 section 10).

7.27xl_cart_to_geod

7.27.1Overview

The **xl_cart_to_geod** CFI function transforms from cartesian to geodetic coordinates.
The user can choose the method for the calculation of the geodetic coordinates by setting the input variable "mode":

- Bowring **iterative** method: This method is more accurate but it provides a poor runtime perfomance. The mode input parameter values are:
  - XL_CALC_POS or XL_CALC_ITER_POS: for the transformation of a position vector.
  - XL_CALC_POS_VEL or XL_CALC_ITER_POS_VEL: for the transformation of the position and the velocity vectors
- Bowring **direct** method: less accurate than the iterative method at the satellite height, but it provides a better runtime perfomance. The mode input parameters values are:
  - XL_CALC_NO_ITER_POS: for the transformation of a position vector.
  - XL_CALC_NO_ITER_POS_VEL: for the transformation of the position and the velocity vectors

The following table shows the difference in accuracy between the two methods:

Method		Max Latitude Error [deg]	Max altitude Error [m]
Iterative	Satellite Height (~700km)	$\sim 10^{-5}$	$\sim 10^{-5}$
	Ground	$\sim 10^{-5}$	$\sim 10^{-5}$
Direct	Satellite Height (~700km)	$\sim 10^{-3}$	$\sim 10^{-3}$
	Ground	$\sim 10^{-5}$	$\sim 10^{-5}$

The difference in performance can be seen in "Runtime Performances" section.

7.27.2Calling interface

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

```
#include <explorer_lib.h>
{
        long mode;
        xl_model_id model_id = {NULL};
        double pos[3], vel[3];
        double lon, lat, h, lon_rate, lat_rate, h_rate;
        long status;
        status = xl_cart_to_geod (&model_id, &mode, pos, vel,
```

```
                                    &lon, &lat,&h,
                                    &lon_rate, &lat_rate, &h_rate);
}
```

7.27.3Input parameters

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

Table 101: Input parameters of xl_cart_to_geod function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
model_id	xl_model_id	-	Model ID	-	-
mode	long*	-	Calculation mode selection	-	Select either: XL_CALC_POS XL_CALC_POS_VEL XL_CALC_ITER_POS XL_CALC_ITER_POS_VEL XL_CALC_NO_ITER_POS XL_CALC_NO_ITER_POS_VEL
pos[3]	double	all	Cartesian position vector (Earth fixed CS)	m	$r > a_{ellipsoid} - b_{ellipsoid}$
vel[3]	double	all	Cartesian velocity vector (Earth fixed CS) Dummy if *mode* is: XL_CALC_POS XL_CALC_ITER_POS XL_CALC_NO_ITER_POS	m/s	-

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

- Calculation mode selection: mode. See current document, section 6.2.

7.27.4Output parameters

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

Table 102: Output parameters of xl_cart_to_geod function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_cart_to_geod	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	-
lon_rate	double *	-	Geocentric longitude rate (Earth fixed CS)	deg/s	-

			Returned only if *mode* is: XL_CALC_POS_VEL XL_CALC_ITER_POS_VEL XL_CALC_NO_ITER_POS_VEL		
lat_rate	double *	-	Geodetic latitude rate (Earth fixed CS) Returned only if *mode* is: XL_CALC_POS_VEL XL_CALC_ITER_POS_VEL XL_CALC_NO_ITER_POS_VEL	deg/s	-
h_rate	double *	-	Geodetic altitude rate (Earth fixed CS) Returned only if *mode* is: XL_CALC_POS_VEL XL_CALC_ITER_POS_VEL XL_CALC_NO_ITER_POS_VEL	m/s	-

7.27.5Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_cart_to_geod** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 **xl_cart_to_geod** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 103: Error messages of xl_cart_to_geod function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Internal computation error # 1	No calculation performed	XL_CFI_CART_GEOD_FRAME_ERR	0
ERR	Input vector out of valid range	No calculation performed	XL_CFI_CART_GEOD_VECTOR_ERR	1
WARN	Calculation mode ID is not correct	Calculation performed. A message informs the user.	XL_CFI_CART_GEOD_MODE_WARN	2
WARN	Geocentric longitude set to 0 deg (ambiguous case)	Calculation performed. A message informs the user.	XL_CFI_CART_GEOD_AMBIGUITY_WARN	3
WARN	Internal computation warning # 1	Calculation performed. A message informs the user.	XL_CFI_CART_GEOD_ACCURACY_WARN	4
WARN	Internal computation warning # 2	Calculation performed. A message informs the user.	XL_CFI_CART_GEOD_ITERATIONS_WARN	5
WARN	Internal computation warning # 3	Calculation performed. A message informs the user.	XL_CFI_CART_GEOD_DEFVAL_WARN	6

7.28xl_kepl_to_cart

7.28.1Overview

The **xl_kepl_to_cart** CFI function transforms from keplerian to cartesian coordinates.

7.28.2Calling interface

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

```
#include <explorer_lib.h>
{
      xl_model_id model_id = {NULL};
      long kepl_mode;
      double kepl_in[6];
      double pos_out[3], vel_out[3];
      long ierr[XL_NUM_ERR_KEPL_CART], status;

      status =    xl_kepl_to_cart (&model_id, &kepl_mode,
                                   kepl_in, pos_out,
                                   vel_out, ierr);

}
```

7.28.3Input parameters

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

Table 104: Input parameters of xl_kepl_to_cart function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
model_id	xl_model_id*	-	Model ID	-	Complete
kepl_mode	long*	-	Flag for selecting: · Mean elements = XL_KEPLER_MEAN · Osculating elements = XL_KEPLER_OSC	-	Complete
kepl_in[6]	double	[0]	Semi-major axis (True of Date CS)	m	>= 0
		[1]	Eccentricity (True of Date CS)	-	[0,1)
		[2]	Inclination (True of Date CS)	deg	[0,180]

		[3]	Right ascension of the ascending node (True of Date CS)	deg	[0,360)
		[4]	Argument of perigee (True of Date CS)	deg	[0,360)
		[5]	Mean anomaly (True of Date CS)	deg	[0,360)

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

- Kepler state vector model: kepl_mode. See section 6.2.

7.28.4Output parameters

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

Table 105: Output parameters of xl_kepl_to_cart function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_kepl_to_cart	long	-	Status flag	-	-
pos_out[3]	double	all	Cartesian position vector (True of Date CS)	m	-
vel_out[3]	double	all	Cartesian velocity vector (True of Date CS)	m/s	-
ierr	long	-	Error vector	-	-

7.28.5Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_kepl_to_cart** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 error vector returned by the **xl_kepl_to_cart** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM])

Table 106: Error messages of xl_kepl_to_cart function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Input semi-major axis <= 0	No calculation performed	XL_CFI_K2C_A_ZERO_ERR	0
ERR	Input eccentricity < 0	No calculation performed	XL_CFI_K2C_E_ZERO_ERR	1
ERR	Input eccentricity > 1	No calculation performed	XL_CFI_K2C_E_ONE_ERR	2
ERR	Internal Error: Error in calling XL_Mean_to_osc	No calculation performed	XL_CFI_K2C_INTERNAL_M2O_ERR	3
ERR	Internal computation error #1	No calculation performed	XL_CFI_K2C_COMPUTATI	4

			ON_ERR	
WARN	Internal Warning: Warning in calling XL_Mean_to_osc	Calculation performed. A message informs the user.	XL_CFI_K2C_INTERNAL_M2O_WARN	5
WARN	Kepler's equations not converged	Calculation performed. A message informs the user.	XL_CFI_K2C_NO_CONVERGED_WARN	6

7.29xl_cart_to_kepl

7.29.1Overview

The **xl_cart_to_kepl** CFI function transforms from cartesian to keplerian coordinates.

7.29.2Calling interface

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

```
#include <explorer_lib.h>
{
      xl_model_id model_id = {NULL};
      long kepl_mode;
      double pos_in[3], vel_in[3];
      double kepl_out[6];
      long ierr[XL_NUM_ERR_CART_KEPL], status;

      status =    xl_cart_to_kepl (&model_id,
                                   pos_in, vel_in, &kepl_mode,
                                   kepl_out, ierr);

}
```

7.29.3Input parameters

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

Table 107: Input parameters of xl_cart_to_kepl function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
model_id	xl_model_id	-	Model ID	-	-
pos_in[3]	double	all	Cartesian position vector (True of Date CS)	m	-
vel_in[3]	double	all	Cartesian velocity vector (True of Date CS)	m/s	-
kepl_mode	long*	-	Flag for selecting: · Mean elements = XL_KEPLER_MEAN · Osculating elements = XL_KEPLER_OSC	-	Complete

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

- Kepler state vector model: kepl_mode. See section 6.2.

7.29.4Output parameters

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

Table 108: Output parameters of xl_cart_to_kepl function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_cart_to_kepl	long	-	Status flag	-	-
kepl_out[6]	double	[0]	Semi-major axis (True of Date CS)	m	>= 0
		[1]	Eccentricity (True of Date CS)	-	[0,1)
		[2]	Inclination (True of Date CS)	deg	[0,180]
		[3]	Right ascension of the ascending node (True of Date CS)	deg	[0,360)
		[4]	Argument of perigee (True of Date CS)	deg	[0,360)
		[5]	Mean anomaly (True of Date CS)	deg	[0,360)
ierr	long	-	Error vector	-	-

7.29.5Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_cart_to_kepl** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 error vector returned by the **xl_cart_to_kepl** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 109: Error messages of xl_cart_to_kepl function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Earth's Mu < 0	No calculation performed	XL_CFI_C2K_MU_ZERO_ERR	0
ERR	Input orbit radius = 0	No calculation performed	XL_CFI_C2K_OR_ZERO_ERR	1
ERR	Input orbit velocity = 0	No calculation performed	XL_CFI_C2K_OV_ZERO_ERR	2
ERR	Semi-major axis undefined	No calculation performed	XL_CFI_C2K_OA_UNDEFINED_ERR	3

ERR	Semi-major axis < 0	No calculation performed	XL_CFI_C2K_OA_ZERO_ERR	4
ERR	Internal computation error #1	No calculation performed	XL_CFI_C2K_COMPUTATION_ERR	5
ERR	Internal Error: Error in calling XL_Osc_to_mean	No calculation performed	XL_CFI_C2K_INTERNAL_O2M_ERR	6
WARN	Inclination = 0 or 180 deg	Calculation performed. A message informs the user.	XL_CFI_C2K_OI_ZERO_WARN	7
WARN	Eccentricity = 0	Calculation performed. A message informs the user.	XL_CFI_C2K_OE_ZERO_WARN	8
WARN	Internal Warning: Warning in calling XL_Osc_to_mean	Calculation performed. A message informs the user.	XL_CFI_C2K_INTERNAL_O2M_WARN	9

7.30xl_cart_to_radec

7.30.1Overview

The **xl_cart_to_radec** CFI function transforms cartesian coordinates to spherical coordinates:

- From equatorial cartesian coordinates to right ascension and declination.

  or

- From galactic cartesian coordinates to galactic longitude and latitude.

7.30.2Calling interface

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

```
#include <explorer_lib.h>
{
      xl_model_id model_id = {NULL};
      long mode, cs_in;
      double pos[3], vel[3];
      double ra, dec, mu_ra, mu_dec, rad_vel, par;
      long ierr[XL_NUM_ERR_CART_RADEC], status;

      status =    xl_cart_to_radec (&model_id, &mode, &cs_in, pos, vel,
                                    &ra, &dec, &mu_ra, &mu_dec,
                                    &rad_vel, &par, ierr);
}
```

7.30.3Input parameters

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

Table 110: Input parameters of xl_cart_to_radec function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
model_id	xl_model_id	-	Model ID	-	-
mode	long*	-	Flag to select transformation, position or position and velocity: • XL_CALC_POS • XL_CALC_POS_VEL	-	Complete

			For galactic coordinates only position can be transformed.		
cs_in	long*	-	Coordinate reference frame for the input vector.	-	All except XL_EF
pos[3]	double	all	Cartesian position vector	m	-
vel[3]	double	all	Cartesian velocity vector	m/s	-

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

- Calculation mode: mode. See section 6.2
- Reference frame: cs_in. See section 6.2

7.30.4Output parameters

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

Table 111: Output parameters of xl_cart_to_radec function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_cart_to_radec	long	-	Status flag	-	-
ra	double	-	Right ascension (or galactic longitude)	rad	[0, 2π)
dec	double	-	Declination (or galactic latitude)	rad	[-π/2,π /2]
mu_ra	double	-	Proper motion in the right ascension	rad/century	-
mu_dec	double	-	Proper motion in the declination	rad/century	-
rad_vel	double	-	Radial velocity	AU/century	-
par	double	-	Parallax	rad	[0, 2π)
ierr	long	-	Error vector	-	-

7.30.5Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_cart_to_radec** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 error vector returned by the **xl_cart_to_radec** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 112: Error messages of xl_cart_to_radec function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Not possible to calculate velocity vector in galactic frame.	No calculation performed	XL_CFI_CART_TO_RADEC_WRONG_INPUT_ERR	0
ERR	Mode input is not an allowed value.	No calculation performed	XL_CFI_CART_TO_RADEC_WRONG_MODE_ERR	1

ERR	cs_in input is not an allowed value.	No calculation performed	XL_CFI_CART_TO_RADEC_WRONG_CS_IN_ERR	2
ERR	The frame's center is not an allowed position input	No calculation performed	XL_CFI_CART_TO_RADEC_WRONG_POSITION_ERR	3

7.31xl_radec_to_cart

7.31.1Overview

The **xl_radec_to_cart** CFI function transforms spherical coordinates to cartesian coordinates:

- From right ascension and declination to equatorial cartesian coordinates.

or

- From galactic longitude and latitude to galactic cartesian coordinates.

7.31.2Calling interface

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

```
#include <explorer_lib.h>
{
       xl_model_id model_id = {NULL};
       long mode, cs_in;
       double pos[3], vel[3];
       double ra, dec, mu_ra, mu_dec, rad_vel, par;
       long ierr[XL_NUM_ERR_RADEC_CART], status;

       status =     xl_radec_to_cart (&model_id, &mode,
                                      &cs_in, &ra, &dec,
                                      &mu_ra, &mu_dec,
                                      &rad_vel, &par,
                                      pos, vel, ierr);
}
```

7.31.3Input parameters

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

Table 113: Input parameters of xl_radec_to_cart function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
model_id	xl_model_id	-	Model ID	-	-
mode	long*	-	Flag to select transformation, position or position and velocity:	-	Complete

			• XL_CALC_POS • XL_CALC_POS_VEL For galactic coordinates only position can be transformed.		
cs_in	long*	-	Coordinate reference frame for the input vector.	-	All except XL_EF
ra	double	-	Right ascension (or galactic longitude)	rad	[0, 2π)
dec	double	-	Declination (or galactic latitude)	rad	[-π/2,π /2]
mu_ra	double	-	Proper motion in the right ascension	rad/century	-
mu_dec	double	-	Proper motion in the declination	rad/century	-
rad_vel	double	-	Radial velocity	AU/century	-
par	double	-	Parallax	rad	[0, 2π)

7.31.4Output parameters

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

Table 114: Output parameters of xl_radec_to_cart function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_radec_to_cart	long	-	Status flag	-	-
pos[3]	double	all	Cartesian position vector	m	-
vel[3]	double	all	Cartesian velocity vector	m/s	-
ierr	long	-	Error vector	-	-

7.31.5Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_radec_to_cart** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 error vector returned by the **xl_radec_to_cart** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 115: Error messages of xl_radec_to_cart function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Not possible to calculate velocity vector in galactic frame.	No calculation performed	XL_CFI_RADEC_TO_CART_WRONG_INPUT_ERR	0
ERR	Mode input is not an allowed value.	No calculation performed	XL_CFI_RADEC_TO_CART_WRONG_MODE_ERR	1
ERR	cs_in input is not an allowed value."	No calculation performed	XL_CFI_RADEC_TO_CART_WRONG_CS_IN_ERR	2
ERR	parallax can't be equal to zero.	No calculation performed	XL_CFI_RADEC_TO_CART_PAR_ERR	3

7.32xl_topocentric_to_ef

7.32.1Overview

The **xl_topocentric_to_ef** CFI function transforms topocentric azimuth and elevation to the Earth Fixed Referece frame.

7.32.2Calling interface

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

```
#include <explorer_lib.h>
{
      xl_model_id model_id = {NULL};
      long   mode, deriv;
      double pos[3], vel[3];
      double azim, elev, range,
             azim_d, elev_d, range_d,
             ef_dir[3], ef_dir_d[3];
      long ierr[XL_NUM_ERR_TOP_TO_EF], status;

      status =    xl_topocentric_to_ef(&model_id, &mode, &deriv, pos,
vel,
                                       &azim, &elev, &range,
                                       &azim_d, &elev_d, &range_d,
                                       ef_dir, ef_dir_d,
                                       ierr);
}
```

7.32.3Input parameters

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

Table 116: Input parameters of xl_topocentric_to_ef function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
model_id	xl_model_id	-	Model ID	-	-
mode	long	-	Flag to indicate if the input coordinates is location or a direction	-	• XL_MODE_FLAG_LOCATION

					• XL_MODE_FLAG _DIRECTION
• deriv	long	-	Flag to indicate if the 1st. derivative has to be computed.	-	• XL_NO_DER • XL_DER_1ST
• pos	double	all	Position of the topocentric CS in the EF CS	m	-
vel	double	all	Velocity of the topocentric CS in the EF CS	m/s	-
azim	double	-	Azimuth	deg	[0, 360)
elev	double	-	Elevation	deg	[-90, +90]
range	double	-	Distance	m	-
azim_d	double	-	Azimuth rate	deg/s	[0, 360)
elev_d	double	-	Elevation rate	deg/s	[-90, +90]
range_d	double	-	Range rate	m/s	-

7.32.4Output parameters

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

Table 117: Output parameters of xl_topocentric_to_ef function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_topocentric_to_ef	long	-	Status flag	-	-
ef_dir	double	all	Cartesian position vector in EF	m	-
ef_dir_d	double	all	Cartesian velocity vector in EF	m/s	-
ierr	long	-	Error vector	-	-

7.32.5Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_topocentric_to_ef** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 error vector returned by the **xl_topocentric_to_ef** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 118: Error messages of xl_topocentric_to_ef function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Wrong for the parameter Location/Direction	No calculation performed	XL_CFI_TOP_TO_EF_WRONG_MODE_FLAG_ERR	0
ERR	Wrong parameter for the derivative	No calculation performed	XL_CFI_TOP_TO_EF_WRONG_DERIV_FLAG_ERR	1

ERR	Could not convert input vector for the topocentric center to geodetic coordinates	No calculation performed	XL_CFI_TOP_TO_EF_CART_TO_GEOD_ERR	2
ERR	Could not get the pointing direction from the input Azimuth and elevation	No calculation performed	XL_CFI_TOP_TO_EF_POINTING_DIR_ERR	3

7.33xl_ef_to_topocentric

7.33.1Overview

The **xl_ef_to_topocentric** CFI function transforms Earth Fixed coordinates to topocentric coordinates for a given ground position.

7.33.2Calling interface

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

```
#include <explorer_lib.h>
{
      xl_model_id model_id = {NULL};
      long   mode, deriv;
      double pos[3], vel[3];
      double azim, elev, range,
             azim_d, elev_d, range_d,
             ef_dir[3], ef_dir_d[3];
      long ierr[XL_NUM_ERR_TOP_TO_EF], status;

      status =    xl_ef_to_topocentric(&model_id,
                                       &mode, &deriv, pos, vel,
                                       ef_dir, ef_dir_d,
                                       &azim, &elev, &range,
                                       &azim_d, &elev_d, &range_d,
                                       ierr);
}
```

7.33.3Input parameters

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

Table 119: Input parameters of xl_ef_to_topocentric function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
model_id	xl_model_id	-	Model ID	-	-
mode	long	-	Flag to indicate if the input coordinates is location or a direction	-	• XL_MODE_FLAG_LOCATION

					• XL_MODE_FLAG _DIRECTION
• deriv	long	-	Flag to indicate if the 1st. derivative has to be computed.	-	• XL_NO_DER • XL_DER_1ST
• pos	double	all	Position of the topocentric CS in the EF CS	m	-
vel	double	all	Velocity of the topocentric CS in the EF CS	m/s	-
ef_dir	double	all	Cartesian position vector in EF	m	-
ef_dir_d	double	all	Cartesian velocity vector in EF	m/s	-

7.33.4Output parameters

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

Table 120: Output parameters of xl_ef_to_topocentric function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_topocentric_to_ef	long	-	Status flag	-	-
azim	double	-	Azimuth	deg	[0, 360)
elev	double	-	Elevation	deg	[-90, +90]
range	double	-	Distance	m	-
azim_d	double	-	Azimuth rate	deg/s	[0, 360)
elev_d	double	-	Elevation rate	deg/s	[-90, +90]
range_d	double	-	Range rate	m/s	-
ierr	long	-	Error vector	-	-

7.33.5Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_ef_to_topocentric** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 error vector returned by the **xl_ef_to_topocentric** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 121: Error messages of xl_ef_to_topocentric function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Wrong for the parameter Location/Direction	No calculation performed	XL_CFI_EF_TO_TOP_WRONG_MODE_FLAG_ERR	0
ERR	Could not convert input vector for the topocentric center to	No calculation performed	XL_CFI_EF_TO_TOP_CART_TO_GEOD_ERR	1

	geodetic coordinates			
ERR	Wrong parameter for the derivative	No calculation performed	XL_CFI_EF_TO_TOP_WRONG_DERIV_FLAG_ERR	2
ERR	Error when computing Azimuth and Elevation	No calculation performed	XL_CFI_EF_TO_TOP_DIR_POINTING_ERR	3

7.34xl_sun

7.34.1Overview

The **xl_sun** CFI function calculates the position and velocity vector of the Sun in the Earth Fixed coordinate system.

7.34.2Calling interface

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

```
#include <explorer_lib.h>
{
       long time_ref;
       double time, sun_pos[3], sun_vel[3];
        xl_model_id model_id = {NULL};
        xl_time_id time_id = {NULL};
       long ierr[XL_NUM_ERR_SUN], status;

       status =     xl_sun(&model_id,
                           &time_id, &time_ref, &time,
                           sun_pos, sun_vel,
                           ierr);

       /* Or, using the run_id */
       long run_id;

       status =  xl_sun_run(&run_id, &time_ref, &time, sun_pos, sun_vel,
                            ierr);
}
```

7.34.3Input parameters

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

Table 122: Input parameters of xl_sun function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
model_id	xl_model_i	-	Model ID	-	-

	d*				
time_id	xl_time_id*	-	Structure that contains the time correlations.	-	-
time_ref	long *	-	Initial time reference ID	-	Any except XL_TIME_UNDEF
time	double*	-	Input time	Decimal days (Processing format)	[-18262.0,36524.0]

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

- Time reference ID: time_ref. See [GEN_SUM].

Note that for the function to work correctly, the time references should be properly initialised before calling the function (see section 4.2 for details).

7.34.4Output parameters

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

Table 123: Output parameters of xl_sun function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_sun	long	-	Status flag	-	-
sun_pos[3]	double	all	Position vector of the Sun in the Earth Fixed CS	m	-
sun_vel[3]	double	all	Velocity vector of the Sun in the Earth Fixed CS	m/s	-
ierr	long	-	Error vector	-	-

7.34.5Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_sun** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 error vector returned by the **xl_sun** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 124: Error messages of xl_sun function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Input time reference ID is not correct	No calculation performed	XL_CFI_SUN_REF_ERR	0
ERR	Input date is out of range	No calculation performed	XL_CFI_SUN_DAY_ERR	1
ERR	Time Reference not initialised	No calculation performed	XL_CFI_SUN_REF_INIT_ERR	2

ERR	Error in calling XL_Sun_PosVel	No calculation performed	XL_CFI_SUN_SUN_ERR	3

7.35xl_moon

7.35.1Overview

The **xl_moon** CFI function calculates the position and velocity vector of the Moon in the Earth Fixed coordinate system.

7.35.2Calling interface

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

```
#include <explorer_lib.h>
{
       long time_ref;
       double time, moon_pos[3], moon_vel[3];
       xl_model_id model_id = {NULL};
       xl_time_id time_id = {NULL};
       long ierr[XL_NUM_ERR_MOON], status;

       status =     xl_moon(&model_id,
                        &time_id, &time_ref, &time,
                        moon_pos, moon_vel,
                        ierr);

       /* Or, using the run_id */
       long run_id;

       status = xl_moon_run(&run_id, &time_ref, &time,
                            moon_pos, moon_vel,
                            ierr);
}
```

7.35.3Input parameters

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

Table 125: Input parameters of xl_moon function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
model_id	xl_model_id*	-	Model ID	-	-
time_id	xl_time_id*	-	Structure that contains the time correlations.	-	-
time_ref	long *	-	Initial time reference ID	-	Any except XL_TIME_UNDEF
time	double*	-	Input time	Decimal days (Processing format)	[-18262.0,36524.0]

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

- Time reference ID: time_ref. See [GEN_SUM].

Note that for the function to work correctly, the time references should be properly initialised before calling the function (see section 4.2. for details).

7.35.4Output parameters

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

Table 126: Output parameters of xl_moon function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_moon	long	-	Status flag	-	-
moon_pos[3]	double	all	Position vector of the Moon in the Earth Fixed CS	m	-
moon_vel[3]	double	all	Velocity vector of the Moon in the Earth Fixed CS	m/s	-
ierr	long	-	Error vector	-	-

7.35.5Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_moon** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 error vector returned by the **xl_moon** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 127: Error messages of xl_moon function

Error	Error message	Cause and impact	Error code	Error

type				No
ERR	Input time reference ID is not correct	No calculation performed	XL_CFI_MOON_REF_ERR	0
ERR	Input date is out of range	No calculation performed	XL_CFI_MOON_DAY_ERR	1
ERR	Time Reference not initialised	No calculation performed	XL_CFI_MOON_REF_INIT_ERR	2
ERR	Error in calling XL_Moon_PosVel	No calculation performed	XL_CFI_MOON_MOON_ERR	3

7.36xl_planet

7.36.1Overview

The **xl_planet** CFI function calculates the position and velocity vector of a planet in the Earth Fixed coordinate system.

7.36.2Calling interface

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

```
#include <explorer_lib.h>
{
      long sat_id, planet, time_ref;
      double time, planet_pos[3], planet_vel[3];
      xl_model_id model_id = {NULL};
      xl_time_id time_id = {NULL};
      long ierr[XL_NUM_ERR_PLANET], status;

      status =    xl_planet(&model_id, &time_id,
                            &planet, &time_ref, &time,
                            planet_pos, planet_vel, ierr);

      /* Or, using the run_id */
      long run_id;

      status =    xl_planet_run(&run_id, &planet, &time_ref, &time,
                                planet_pos, planet_vel, ierr);
}
```

7.36.3Input parameters

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

Table 128: Input parameters of xl_planet function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
model_id	xl_model_id*	-	Model ID	-	-
time_id	xl_time_id*	-	Structure that contains	-	-

			the time correlations.		
planet	long *	-	Planet ID	-	Complete
time_ref	long *	-	Initial time reference ID	-	Any except XL_TIME_UNDEF
time	double*	-	Input time	Decimal days (Processing format)	[-18262.0,36524.0]

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

- Time reference ID: time_ref. See [GEN_SUM].
- Planet ID: planet. Current document, section 6.2.

Note that for the function to work correctly, the time references should be properly initialised before calling the function (see section 4.2 for details).

7.36.4Output parameters

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

Table 129: Output parameters of xl_planet function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_planet	long	-	Status flag	-	-
planet_pos[3]	double	all	Position vector of the Planet in the Earth Fixed coordinate system	m	-
planet_vel[3]	double	all	Velocity vector of the Planet in the Earth Fixed coordinate system	m/s	-
ierr	long	-	Error vector	-	-

7.36.5Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_planet** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 error vector returned by the **xl_planet** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 130: Error messages of xl_planet function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Input time reference ID is not correct	No calculation performed	XL_CFI_PLANET_REF_ERR	0

ERR	Input date is out of range	No calculation performed	XL_CFI_PLANET_DAY_ERR	1
ERR	Time Reference not initialised	No calculation performed	XL_CFI_PLANET_REF_INIT_ERR	2
ERR	Planet code is not correct	No calculation performed	XL_CFI_PLANET_PLANET_ERR	3
WARN	Internal Warning: XL_Planets solution didn't converge	Calculation performed. A message informs the user.	XL_CFI_PLANET_CONV_WARN	4

7.37xl_star_radec

7.37.1Overview

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

7.37.2Calling interface

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

```
#include <explorer_lib.h>
{
       lond time_ref;
       double time, ra0, dec0, mu_ra, mu_dec;
       double rad_vel, par, ra, dec;
       xl_time_id time_id = {NULL};
       xl_model_id model_id = {NULL};
       long ierr[XL_NUM_ERR_STAR], status;

       status =     xl_star_radec(&model_id, &time_id,
                                  &time_ref, &time, &ra0, &dec0,
                                  &mu_ra, &mu_dec, &rad_vel, &par,
                                  &ra, &dec, ierr);

       /* Or, using the run_id */
       long run_id;

      status = xl_star_radec_run(&run_id, &time_ref, &time, &ra0, &dec0,
                                  &mu_ra, &mu_dec, &rad_vel, &par,
                                  &ra, &dec, ierr);
}
```

7.37.3Input parameters

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

Table 131: Input parameters of xl_star_radec function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
model_id	xl_model_id*	-	Model ID	-	-
time_id	xl_time_id*	-	Structure that contains the time correlations.	-	-
time_ref	long *	-	Initial time reference ID	-	Any except XL_TIME_UNDEF
time	double*	-	Input time	Decimal days (Processing format)	[-18262.0,36524.0]
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	-

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

- Time reference ID: time_ref. See [GEN_SUM].

Note that for the function to work correctly, the time references should be properly initialised before calling the function (see section 4.2 for details).

7.37.4Output parameters

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

Table 132: Output parameters of xl_star_radec function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_star_radec	long	-	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]
ierr	long	-	Error vector	-	-

7.37.5Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_star_radec** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 error vector returned by the **xl_star_radec** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 133: Error messages of xl_star_radec function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Input time reference ID is not correct	No calculation performed	XL_CFI_STAR_RADEC_REF_ERR	0
ERR	Input date is out of range	No calculation performed	XL_CFI_STAR_RADEC_DAY_ERR	1
ERR	Time Reference not initialised	No calculation performed	XL_CFI_STAR_RADEC_REF_INIT_ERR	2
ERR	Error in calling XL_Star	No calculation performed	XL_CFI_STAR_RADEC_STAR_ERR	3
ERR	Error in calling XL_Dir_Pointing	No calculation performed	XL_CFI_STAR_RADEC_DIRPOINT_ERR	4
WARN	Warning in calling XL_Dir_Pointing	Calculation performed. A message informs the user.	XL_CFI_STAR_RADEC_DIRPOINT_WARN	5

The declination is not checked, so in case it does not satisfy its allowed range it may result in raising an internal error (see section 10).

7.38xl_star_catalog

7.38.1Overview

The **xl_star_catalog** CFI function calculates the right ascension and declination of a star in a selected star catalogue.

7.38.2Calling interface

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

```
#include <explorer_lib.h>
{
       lond time_ref, catalog_in, cs_out, mode;
       double time, ra0, dec0, mu_ra0, mu_dec0;
       double rad_vel0, par0, ra, dec;
       xl_model_id model_id = {NULL};
       xl_time_id time_id = {NULL};
       long ierr[XL_NUM_ERR_STAR_CATALOG], status;

       status =     xl_star_catalog(&model_id, &time_id,
                                    &time_ref, &time, &mode,
                                    &catalog_in, &catalog_out, &ra0, &dec0,
                                    &mu_ra0, &mu_dec0, &rad_vel0, &par0,
                                    &ra, &dec, ierr);
       /* Or, using the run_id */
       long run_id;
       status = xl_star_catalog_run(&run_id, &time_ref, &time, &mode,
                                    &catalog_in, &catalog_out, &ra0, &dec0,
                                    &mu_ra0, &mu_dec0, &rad_vel0, &par0,
                                    &ra, &dec, ierr);
}
```

7.38.3Input parameters

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

Table 134: Input parameters of xl_star_catalog function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
model_id	xl_model_id*	-	Model ID.	-	-
time_id	xl_time_id*	-	Structure that contains the time correlations.	-	-
time_ref	long *	-	Initial time reference ID	-	Any except XL_TIME_UNDEF
time	double*	-	Input time	Decimal days (Processing format)	[-18262.0,36524.0]
mode	long*	-			
catalog_in	long*	-	Input star catalog	-	All
catalog_out	long*	-	Output coordinate frame	-	All
ra0	double *	-	Right ascension of the star in the input catalog	rad	[0,2π)
dec0	double *	-	Declination of the star in the input catalog	rad	[-π/2,π/2]
mu_ra0	double *	-	Proper motion in the right ascension in the input catalog	rad/century	-
mu_dec0	double *	-	Proper motion in the declination in the input catalog	rad/century	-
rad_vel0	double *	-	Radial velocity of the star in the input catalog	AU/century	-
par0	double *	-	Parallax of the star in the input catalog	rad	-

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

- Time reference ID: time_ref. See [GEN_SUM].
- Star catalog ID: catalog_in.See section 6.2
- Reference frame: cs_out. See section 6.2

Note that for the function to work correctly, the time references should be properly initialised before calling the function (see section 4.2 for details).

7.38.4Output parameters

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

Table 135: Output parameters of xl_star_catalog function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_star_catalog	long	-	Status flag	-	-
ra	double *	-	Right ascension of the star at specified time in the out_cs reference frame.	rad	[0,2π)
dec	double *	-	Declination of the star at specified time in the out_cs reference frame.	rad	[-π/2, π/2]

ierr	long	-	Error vector	-	-

7.38.5Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_star_catalog** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 error vector returned by the **xl_star_catalog** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 136: Error messages of xl_star_catalog function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Wrong input catalog	No calculation performed	XL_CFI_STAR_CATALOG_WRONG_INPUT_CATALOG_ERR	0
ERR	Wrong output catalog	No calculation performed	XL_CFI_STAR_CATALOG_WRONG_OUTPUT_CATALOG_ERR	1
ERR	Error in xl_star_radec	No calculation performed	XL_CFI_STAR_CATALOG_STAR_RADEC_ERR	2
ERR	Error when converting from FK4 to FK5	No calculation performed	XL_CFI_STAR_CATALOG_FK4_TO_FK5_ERR	3
ERR	Error in xl_radec_to_cart	No calculation performed	XL_CFI_STAR_CATALOG_RADEC_TO_CART_ERR	4
ERR	Error in xl_change_coordinate_cs	No calculation performed	XL_CFI_STAR_CATALOG_CHANGE_CART_CS_ERR	5
ERR	Error in xl_cart_to_radec	No calculation performed	XL_CFI_STAR_CATALOG_CART_TO_RADEC_ERR	6

7.39xl_geod_distance

7.39.1Overview

The **xl_geod_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.

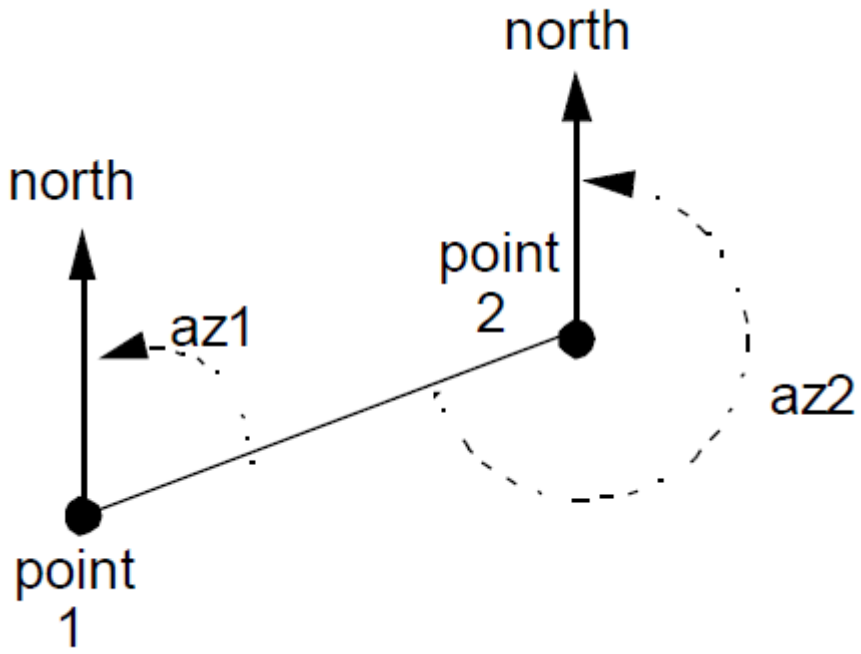


Figure 3: Azimuth figures returned by xl_geod_distance function

7.39.2Calling interface

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

```
#include <explorer_lib.h>
{
      xl_model_id model_id = {NULL};
      double lon1, lat1, lon2, lat2, h;
      double distance, az_1_to_2, az_2_to_1;
      long status;
      status =    xl_geod_distance (&model_id,
                                    &lon1, &lat1, &lon2, &lat2, &h,
                                    &distance, &az_1_to_2,
```

```
                                        &az_2_to_1);
}
```

7.39.3Input parameters

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

Table 137: Input parameters of xl_geod_distance function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
model_id	xl_model_i d	-	Model ID	-	-
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 second input point (Earth fixed CS)	deg	[-90,90]
h	double *	-	Geodetic altitude of both input points (Earth fixed CS)	m	h >= $-b_{WGS}$ (satellite ID dependent)

7.39.4lOutput parameters

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

Table 138: Output parameters of xl_geod_distance function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_geod_distance	long	-	Extended status flag	-	-
distance	double *	-	Geodesic distance between the two input points (Earth fixed CS)	m	>= 0
az_1_to_2	double *	-	Azimuth of the geodesic line from point 1 to point 2 (Topocentric CS)	deg	[0,360)
az_2_to_1	double *	-	Azimuth of the geodesic line from point 2 to point 1 (Topocentric CS) Note that az2 = az1 + 180 approximately	deg	>= 0 < 360

7.39.5Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_geod_distance** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 **xl_geod_distance** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 139: Error messages of xl_geod_distance function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Different altitudes in the two points	No calculation performed	XL_CFI_GEOD_DIST_ALTITUDE_ERR	0
ERR	Calculation not performed in XL_Geo_Car	No calculation performed	XL_CFI_GEOD_DIST_GEO_CAR_ERR	1
ERR	Calculation not performed in XL_Pt_Dir_Range	No calculation performed	XL_CFI_GEOD_DIST_DIR_RANGE_ERR	2
ERR	No solution returned by XL_Dir_Pointing	No calculation performed	XL_CFI_GEOD_DIST_DIR_POINTING_ERR	3
WARN	Antipodal points.Two possible azimuth values (0 or 180).Selected value is 0.0 deg	Calculation performed. A message informs the user.	XL_CFI_GEOD_DIST_ANTIPODAL_POINTS_WARN	4
WARN	Default values returned by XL_Dir_Pointing	Calculation performed. A message informs the user.	XL_CFI_GEOD_DIST_DIR_POINTING_WARN	5

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 section 10).

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.

7.40 xl_time_get_leap_second_info

7.40.1Overview

The **xl_time_get_leap_second_info** CFI function retrieves the leap second location (if any) in the initialised time range.

In order to avoid ambiguities the instant of Leap Second insertion is given both as the instant just before insertion (i.e. when the LS start) and the instant just after insertion (i.e. when the LS ends).

As an example, in the case of the (positive) LS inserted on January 1st, 1999, the function would return (if `ascii_id_out = XL_ASCII_STD_REF_MICROSEC`):

```
leap_flag = 1
ascii_utc_time_before_leap = UTC=1998-12-31_23:59:60.000000
ascii_utc_time_after_leap = UTC=1999-01-01_00:00:00.000000
```

In the case of a negative LS, inserted as an example on January 1st, 2009, the function would return (if `ascii_id_out = XL_ASCII_STD_REF_MICROSEC`):

```
leap_flag = -1
ascii_utc_time_before_leap = UTC=2008-12-31_23:59:58.000000
ascii_utc_time_after_leap = UTC=2009-01-01_00:00:00.000000
```

Note that, if the time correlations where initalised with an Orbit Scenario File, LS could be wrongly calculated (see section 7.1.).

7.40.2Calling interface

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

```
#include <explorer_lib.h>
{
      long ascii_id_out, leap_flag;
      char ascii_utc_time_before_leap[XL_TIME_ASCII_DIM_MAX];
      char ascii_utc_time_after_leap[XL_TIME_ASCII_DIM_MAX]
       xl_time_id time_id = {NULL};
      long ierr[XL_NUM_ERR_LEAP_INFO], status;

      status = xl_time_get_leap_second_info(&time_id, &ascii_id_out,
                           &leap_flag,      ascii_utc_time_before_leap,
                           ascii_utc_time_after_leap, ierr);

      /* Or, using the run_id */
      long run_id;
```

```
        status = xl_time_get_leap_second_info_run(&run_id, &ascii_id_out,
                                &leap_flag,      ascii_utc_time_before_leap,
                                ascii_utc_time_after_leap, ierr);
}
```

The `XL_TIME_ASCII_DIM_MAX` and `XL_NUM_ERR_LEAP_INFO` constants are defined in the file *explorer_lib.h.*

7.40.3Input parameters

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

Table 140: Input parameters of xl_time_get_leap_second_info function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
time_id	xl_time_id*	-	Structure that contains the time correlations.	-	-
ascii_id_out	long *	-	ASCII format ID for output	-	Complete

It is possible to use enumeration values rather than integer values for the input argument:

- ASCII format ID: ascii_id_out. Current document, section 6.2.

7.40.4Output parameters

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

Table 141: Output parameters of xl_time_get_leap_second_info function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_time_get_leap_second_info	long	-	Status flag	-	-
leap_flag	long *	-	Flag for leap second presence within time initialization data	-	-1 = Negative Leap Second (a LS has been removed) (very rare case) 0 = No leap second within initialization data +1 = Positive Leap Second (a LS has been added) (usual case)
ascii_utc_time_before_leap	char	See Table 4 and Table 5	UTC time just before leap second insertion (dummy if leap_flag=0)	See Table 4 and Table 5	See Table 4 and Table 5
ascii_utc_time_after_leap	char	See Table 4 and Table 5	UTC time just after leap second insertion (dummy if leap_flag=0)	See Table 4 and Table 5	See Table 4 and Table 5
ierr	long	-	Error vector	-	-

Note that if more than one leap second is contained within the time initialization data for the selected satellite, only the last (most recent) one is returned.

No more than one leap second is likely to be found in the data, unless the range of time initializazion span more than one year (a total of 23 leap seconds have been inserted until 2002, since the system was introduced in 1972).

7.40.5Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_time_get_leap_second_info** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 by translating the error vector returned by the **xl_time_get_leap_second_info** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 142: Error messages of xl_time_get_leap_second_info function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Output ascii format ID is not correct	No calculation performed	XL_CFI_TIME_LEAP_SECOND_ASCII_OUT_ERR	0
ERR	Satellite ID and output format ID are not compatible	No calculation performed	XL_CFI_TIME_LEAP_SECOND_COMP_OUT_ERR	1
ERR	Error in adding times in Processing format	No calculation performed	XL_CFI_TIME_LEAP_SECOND_ADD_ERR	2
ERR	Error in converting from Processing to ASCII format	No calculation performed	XL_CFI_TIME_LEAP_SECOND_P2A_ERR	3
WARN	Time Reference not initialised	No calculation performed A message informs the user.	XL_CFI_TIME_LEAP_SECOND_TIME_REF_INIT_WARN	4

7.41 xl_euler_to_matrix

7.41.1Overview

The **xl_euler_to_matrix** CFI function computes the rotation matrix equivalent to apply the three consecutive rotation through the given Euler angles. In other words, the result of multiplying the matrix to a vector is the same that applying the Euler rotations to the vector.

The rotation of a vector through the Euler angles is defined as three rotations of the reference frame:

1. Rotation around -Ys over a roll angle h
2. Rotation around -X1s (i.e the rotated Xs) over a pitch angle x
3. Rotation around +Z2s (i.e the rotated Z1s) over a yaw angle z.

Next drawing depicts the three rotations:

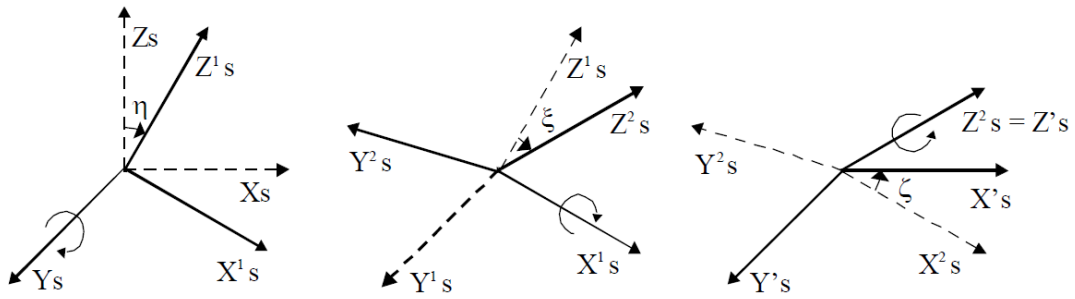


Figure 4: Euler Angles

Note on matrix notation:

If XYZ are the axes of the original reference frame, and X'Y'Z' are the axes of the rotated frame, the rows of the rotation matrix are respectively X, Y and Z axes expressed in X'Y'Z' system.

In the C representation, M[0][], M[1][], M[2][] are respectively 1st, 2nd and 3rd row of a rotation matrix M.

The rotation matrix M satisfies the following equivalence:

$$\mathbf{V} = M \cdot \mathbf{V'}$$

where **V'** is a vector expressed in the X'Y'Z' reference system and **V** is expressed in the XYZ reference system.

7.41.2Calling interface

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

```
#include <explorer_lib.h>
{
      double angles[3];
      double matrix[3][3];
      long ierr[XL_NUM_ERR_EULER_TO_MATRIX], status;

      status =    xl_euler_to_matrix       (angles, matrix, ierr);
}
```

The XL_NUM_ERR_EULER_TO_MATRIX constant is defined in the file *explorer_lib.h.*

7.41.3Input parameters

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

Table 143: Input parameters of xl_euler_to_matrix function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
angles	double[3]	[0]	Pitch angle	degrees	-
		[1]	Roll angle		
		[2]	Yaw angle		

7.41.4Output parameters

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

Table 144: Output parameters of xl_euler_to_matrix function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_euler_to_matrix	long	-	Status flag	-	-
matrix	double [3][3]	All	Rotation matrix equivalent to the Euler angles	-	-
ierr	long	-	Error vector	-	-

7.41.5Warnings and errors

No errors have been envisaged for this function.

7.42xl_matrix_to_euler

7.42.1Overview

The **xl_matrix_to_euler** CFI function computes the Euler angles (see section 7.41.1) equivalent to the input rotation matrix (the matrix is checked to be orthonormal; if not, an error is returned). This function is the inverse of **xl_euler_to_matrix**.

The transformation from a rotation matrix to Euler angles is not unique, there are two sets of angles that lead to the same rotation matrix. More precisely, the rotation given by (*pitch, roll, yaw*) is equivalent to (*180-pitch, 180+roll, 180+yaw*). Of the two possible solutions, this function chooses the one in which the *pitch* angle is between -90º and +90º (or cos(*pitch*) > 0)

Another indetermination happens when the pitch angle is ±90. In this case, the values for roll and yaw depends on each other. In this case the function returns a warning (section 7.42.5) and a solution is returned for which the yaw angle is set to 0.**Note on matrix notation:**

If XYZ are the axes of the original reference frame, and X'Y'Z' are the axes of the rotated frame, the rows of the rotation matrix are respectively X, Y and Z axes expressed in X'Y'Z' system.

In the C representation, M[0][], M[1][], M[2][] are respectively 1st, 2nd and 3rd row of a rotation matrix M.

The rotation matrix M satisfies the following equivalence:

$$\mathbf{V} = M \cdot \mathbf{V'}$$

where **V'** is a vector expressed in the X'Y'Z' reference system and **V** is expressed in the XYZ reference system.

7.42.2Calling interface

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

```
#include <explorer_lib.h>
{
       double angles[3];
       double matrix[3][3];
       long ierr[XL_NUM_ERR_MATRIX_TO_EULER], status;

       status =     xl_matrix_to_euler       (matrix, angles, ierr);
}
```

The `XL_NUM_ERR_MATRIX_TO_EULER` constant is defined in the file *explorer_lib.h*.

7.42.3Input parameters

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

Table 145: Input parameters of xl_matrix_to_euler function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
matrix	double [3][3]	All	Rotation matrix	-	-

7.42.4Output parameters

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

Table 146: Output parameters of xl_matrix_to_euler function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_matrix_to_euler	long	-	Status flag	-	-
angles	double[3]	[0]	Pitch angle	degrees	[-180, 180]
		[1]	Roll angle		
		[2]	Yaw angle		
ierr	long	-	Error vector	-	-

7.42.5Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_matrix_to_euler** CFI function after translating the returned error vector into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 by translating the error vector returned by the **xl_matrix_to_euler** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 147: Error messages of xl_matrix_to_euler function

Error type	Error message	Cause and impact	Error code	Error No
ERR	The matrix does not define a rotation	No calculation performed	XL_CFI_MATRIX_TO_EULER_WRONG_MATRIX_ERR	0
WARN	Ambiguity in angles determination	Calculation performed. The roll and yaw angles are indetermined, so the yaw is set to zero and the roll is computed as if the yaw were 0. In whatever case the three angles are equivalent to the rotation matrix. This situation happens when the pitch angle is 90º or -90º.	XL_CFI_MATRIX_TO_EULER_ANGLES_UNDEFINED_WARN	1

ERR	The matrix is not orthonormal	No calculation performed. The CFI performs a check, with a tolerance of 10^{-6}, that the product of the input matrix and its transposed is the unitary matrix	XL_CFI_MATRIX_TO_EULER_ORTHONORMAL_ERR	2

7.43xl_position_on_orbit

7.43.1Overview

The **xl_position_on_orbit** CFI function calculates the angle describing the position of the satellite within the orbit, using as input a Cartesian orbit state vector in EF. This angle is defined as the angle between the satellite position and the intersection of the orbital plane with a reference plane (the reference plane is the equator in GM2000, ToD or EF CS).

7.43.2 Calling interface

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

```
#include <explorer_lib.h>
{
      long angle_type, time_ref, deriv;
      double time, pos[3], vel[3], acc[3],
      double angle, angle_rate, angle_rate_rate;
       xl_time_id time_id = {NULL};
      long status, ierr[XL_NUM_ERR_POSITION_ON_ORBIT];

      status = xl_position_on_orbit(&model_id,
                                    &time_id,
                                    &angle_type,
                                    &time_ref, &time,
                                    pos, vel, acc, &deriv,
                                    &angle, &angle_rate,
                                    &angle_rate_rate,
                                    ierr);
      /* Or, using the run_id */
      long run_id;
      status = xl_position_on_orbit_run(&run_id,
                                        &angle_type,
                                        &time_ref, &time,
                                        pos, vel, acc, &deriv,
                                        &angle, &angle_rate,
                                        &angle_rate_rate,
                                        ierr);
```

}

7.43.3Input parameters

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

Table 148: Input parameters of xl_position_on_orbit function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
model_id	xl_model_id *	-	Model ID	-	-
time_id	xl_time_id*	-	Structure that contains the time correlations.	-	-
angle_type	long*	-	Type of angle. It defines the reference plane.	-	XL_ANGLE_TYPE_TRUE_LAT_TOD XL_ANGLE_TYPE_TRUE_LAT_GM2000 XL_ANGLE_TYPE_TRUE_LAT_EF
time_ref	long*	-	Time reference ID	-	Complete
time	double*	-	Reference time	Decimal days (Processing format)	[-18262.0,36524.0]
pos	double[3]	all	Satellite position vector (Earth Fixed CS)	m	-
vel	double[3]	all	Satellite velocity vector (Earth Fixed CS)	m/s	-
acc	double[3]	all	Satellite acceleration vector (Earth Fixed CS)	m/s2	-
deriv	long *	-	Derivative ID	-	Allowed values: (0) XL_NO_DER (1) XL_DER_1ST (2) XL_DER_2ND

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

- Time reference ID: time_ref.

7.43.4Output parameters

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

Table 149: Output parameters of xl_position_on_orbit function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
angle	double*	-	Angle describing the position in the orbit	deg	
angle_rate	double*	-	Angle describing the position in the orbit-rate	deg/s	-

angle_rate_rate	double*	-	Angle describing the position in the orbit-rate-rate	deg/s2	-
ierr[XL_NUM_ERR_POSITION_ON_ORBIT]	long	all	Status vector	-	-

7.43.5Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_position_on_orbit** CFI function after translating the returned status vector into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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, mainly on the results vector.

The table is completed by the error code and value. These error codes can be obtained translating the status vector returned by the **xl_position_on_orbit** CFI function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 150: Error messages of xl_position_on_orbit function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Angle type is not valid	No calculation performed	XL_CFI_POSITION_ON_ORBIT_ANGLE_TYPE_ERR	0
ERR	Error occured during call to xl_change_cart_cs	No calculation performed	XL_CFI_POSITION_ON_ORBIT_CHANGE_CART_CS_ERR	1
ERR	Error occured during call to XL_True_Lat	No calculation performed	XL_CFI_POSITION_ON_ORBIT_TRUE_LAT_ERR	2
ERR	Position and velocity are parallel	No calculation performed	XL_CFI_POSITION_ON_ORBIT_PARALLEL_POS_VEL_ERR	3
ERR	Orbit is equatorial	No calculation performed	XL_CFI_POSITION_ON_ORBIT_EQUATORIAL_ORBIT_ERR	4

7.44xl_get_rotation_angles

7.44.1Overview

The **xl_get_rotation_angles** CFI function calculates the rotation angles between two sets of orthonormal right-handed unit vectors expressed wrt an identical coordinate frame.

7.44.2Calling interface

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

```
#include <explorer_lib.h>
{
      double xs_initial[3], ys_initial[3], zs_initial[3];
      double xs_final[3], ys_final[3], zs_final[3];
      double ang[3];
      long ierr[XL_NUM_ERR_GET_ROTATION_ANGLES], status;
      status =    xl_get_rotation_angles (xs_initial, ys_initial,

                        zs_initial, xs_final, ys_final, zs_final,
                        ang, ierr);
}
```

The `XL_NUM_ERR_GET_ROTATION_ANGLES` constant is defined in the file *explorer_lib.h*.

7.44.3Input parameters

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

Table 151: Input parameters of xl_get_rotation_angles function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xs_initial[3]	double	all	Unitary direction vector along the X-axes of the initial attitude frame (Coordinate System)	-	-
ys_initial[3]	double	all	Unitary direction vector along the Y-axes of the initial attitude frame (Coordinate System)	-	-
zs_initial[3]	double	all	Unitary direction vector along the Z-axes of the initial	-	-

			attitude frame (Coordinate System)		
xs_final[3]	double	all	Unitary direction vector along the X-axes of the final attitude frame (Coordinate System)	-	-
ys_final[3]	double	all	Unitary direction vector along the Y-axes of the final attitude frame (Coordinate System)	-	-
zs_final[3]	double	all	Unitary direction vector along the Z-axes of the final attitude frame (Coordinate System)	-	-

7.44.4Output parameters

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

Table 152: Output parameters of xl_get_rotation_angles function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
ang[3]	double	[0]	Pitch angle between initial and final Attitude Frames	deg	[-180,180)
		[1]	Roll angle between initial and final Attitude Frames	deg	[-180,180)
		[2]	Yaw angle between initial and final Attitude Frames	deg	[-180,180)
ierr	long	-	Error vector	-	-

7.44.5Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_get_rotation_angles** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 **xl_get_rotation_angles** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM])

Table 153: Error messages of xl_get_rotation_angles function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Input vectors are not orthogonal	No calculation performed. The CFI performs a check, with a tolerance of 10^{-6}, that the internal product of the	XL_CFI_GET_ROTATION_ANGLES_NO_ORTHOGONAL_ERR	0

		input vectors is zero.		
ERR	Error occured during call to function XL_CS_Rotation	No calculation performed	XL_CFI_GET_ROTATION_ANGLES_CS_ROTATION_ERR	1

7.45xl_get_rotated_vectors

7.45.1Overview

The **xl_get_rotated_vectors** CFI function calculates the rotated unit vectors given a set of unit vectors and the rotation angles expressed wrt an identical coordinate frame.

7.45.2Calling interface

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

```
#include <explorer_lib.h>
{
      double xs_initial[3], ys_initial[3], zs_initial[3];
      double xs_final[3], ys_final[3], zs_final[3];
      double ang[3];
      long ierr[XL_NUM_ERR_GET_ROTATED_VECTORS], status;
      status =    xl_get_rotated_vectors (xs_initial, ys_initial,
                        zs_initial, ang, xs_final, ys_final,
                        zs_final, ierr);
}
```

The `XL_NUM_ERR_GET_ROTATED_VECTORS` constant is defined in the file *explorer_lib.h*.

7.45.3Input parameters

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

Table 154: Input parameters of xl_get_rotated_vectors function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xs_initial[3]	double	all	Unitary direction vector along the X-axes of the initial attitude frame (Coordinate System)	-	-
ys_initial[3]	double	all	Unitary direction vector along the Y-axes of the initial attitude frame (Coordinate System)	-	-
zs_initial[3]	double	all	Unitary direction vector along the Z-axes of the initial attitude frame	-	-

			(Coordinate System)		
ang[3]	double	[0]	Pitch angle between initial and final Attitude Frames	deg	[-180,180)
		[1]	Roll angle between initial and final Attitude Frames	deg	[-180,180)
		[2]	Yaw angle between initial and final Attitude Frames	deg	[-180,180)

7.45.4Output parameters

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

Table 155: Output parameters of xl_get_rotated_vectors function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xs_final[3]	double	all	Unitary direction vector along the X-axes of the rotated attitude frame (Coordinate System)	-	-
ys_final[3]	double	all	Unitary direction vector along the Y-axes of the rotated attitude frame (Coordinate System)	-	-
zs_final[3]	double	all	Unitary direction vector along the Z-axes of the rotated attitude frame (Coordinate System)	-	-
ierr	long	-	Error vector	-	-

7.45.5Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_get_rotated_vectors**CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 **xl_get_rotated_vectors** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 156: Error messages of xl_get_rotated_vectors function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Input vectors are not orthogonal	No calculation performed. The CFI performs a check, with a tolerance of 10^{-6}, that the internal product of the input vectors is zero.	XL_CFI_GET_ROTATED_VECTORS_NO_ORTHOGONAL_ERR	1

ERR	Error occurred during call to function XL_Rotate_CS	No calculation performed	XL_CFI_GET_ROTATED_VECTORS_ROTATE_CS_ERR	2

7.46xl_quaternions_to_vectors

7.46.1Overview

The **xl_quaternions_to_vectors** CFI function calculates the orthonormal unit vectors from a given set of quaternions.

7.46.2Calling interface

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

```
#include <explorer_lib.h>
{

        double quaternions[4];
        double ux_vec[3], uy_vec[3], uz_vec[3];
        long ierr[XL_NUM_ERR_QUATERNIONS_TO_VEC], status;

        status =     xl_quaternions_to_vectors (quaternions,
                                ux_vec, uy_vec, uz_vec, ierr);
}
```

The `XL_NUM_ERR_QUATERNIONS_TO_VECTORS` constant is defined in the file *explorer_lib.h.*

7.46.3Input parameters

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

Table 157: Input parameters of xl_quaternions_to_vectors function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
quaternions	double[4]	-	Quaternions	-	-

7.46.4Output parameters

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

Table 158: Output parameters of xl_quaternions_to_vectors function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
ux_vec[3]	double	all	Unitary direction vector along the X-axes of the coordinate or attitude frame	-	-
uy_vec[3]	double	all	Unitary direction vector along the Y-axes of the coordinate or attitude frame	-	-
uz_vec[3]	double	all	Unitary direction vector along the Z-axes of the rcoordinate or attitude frame	-	-
ierr	long	-	Error vector	-	-

7.46.5Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_quaternions_to_vectors**CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 **xl_quaternions_to_vectors** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 159: Error messages of xl_quaternions_to_vectors function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Wrong input quaternion. The module is different from 1	No calculation performed	XL_CFI_QUATERNIONS_TO_VEC_WRONG_INPUT_ERR	0

7.47xl_vectors_to_quaternions

7.47.1Overview

The **xl_vectors_to_quaternions** CFI function calculates the set of quaternions that correspond to a set of orthonormal unit vectors.

7.47.2Calling interface

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

```
#include <explorer_lib.h>
{
      double quaternions[4];
      double ux_vec[3], uy_vec[3], uz_vec[3];
      long ierr[XL_NUM_ERR_VEC_TO_QUATERNIONS], status;

      status =    xl_vectors_to_quaternions (ux_vec, uy_vec, uz_vec,
                        quaternions, ierr);
}
```

The `XL_NUM_ERR_VECTORS_TO_QUATERNIONS` constant is defined in the file *explorer_lib.h.*

7.47.3Input parameters

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

Table 160: Input parameters of xl_vectors_to_quaternions function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
ux_vec[3]	double	all	Unitary direction vector along the X-axes of the coordinate or attitude frame	-	-
uy_vec[3]	double	all	Unitary direction vector along the Y-axes of the coordinate or attitude frame	-	-
uz_vec[3]	double	all	Unitary direction vector along the Z-axes of the coordinate or attitude frame	-	-

7.47.4Output parameters

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

Table 161: Output parameters of xl_vectors_to_quaternions function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
quaternions	double[4]	-	Quaternions	-	-
ierr	long	-	Error vector	-	-

7.47.5Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_vectors_to_quaternions** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 **xl_vectors_to_quaternions** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 162: Error messages of xl_vectors_to_quaternions function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Wrong input vectors.	No calculation performed	XL_CFI_VEC_TO_QUATERNIONS_WRONG_INPUT_ERR	0
ERR	Input vectors are not orthonormal	No calculation performed. The CFI performs a check, with a tolerance of 10^{-6}, that the internal product of the input vectors is zero.	XL_CFI_VEC_TO_QUATERNIONS_ORTHONORMAL_ERR	1

7.48xl_default_sat_init

7.48.1Overview

The **xl_default_sat_init** CFI function initializes a default satellite from a satellite configuration file (see [D_H_SUM]). This operation is needed whenever a default satellite is to be used for the first time, otherwise the satellite will not be recognized.

When the satellite is initialized, the function returns the satellite identifier (the sat_id). The sat_id cannot be chosen by the user, as the program will give the first available satellite if there is any. In order that a sat_id number can be used again for another initialization, it has to be freed by calling to the CFI function **xl_default_sat_close**.

Important note: Some parameters in the configuration file should be within the following ranges:

- nominal semimajor axis (a) >0
- nominal inclination (i): 0 deg < i < 180 deg
- nominal eccentricity (e): 10-6 deg < e < 1

7.48.2Calling interface

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

```
#include <explorer_lib.h>
{
      long sat_id;
      char *conf_file;
      long ierr[XL_NUM_ERR_DEFAULT_SAT_INIT];
      long status;

      status = xl_default_sat_init(&sat_id, conf_file, ierr);
}
```

7.48.3Input parameters

The **xl_default_sat_init** function has the following input parameters:

Table 163: Input parameters of xl_default_sat_init function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
conf_file	char*	-	Path and name for the Satellite Configuration File (see [D_H_SUM] for further details about the configuration file).	-	-

7.48.4Output parameters

The output parameters of the **xl_default_sat_init** function are:

Table 164: Output parameters of xl_default_sat_init function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
sat_id	long*	-	Satellite ID The value is asigned automatically if there is an available satellite.	-	From XL_SAT_DEFAULT to XL_SAT_DEFAULT9
ierr	long*	all	Error status flags	-	

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

- Satellite ID: sat_id. See [GEN_SUM].

7.48.5Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_default_sat_init** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 **xl_default_sat_init** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 165: Error messages of xl_default_sat_init function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Default satellite ID is not correct	The satellite identification number does not belong to a default satellite. No computation performed	XL_CFI_DEFAULT_SAT_INIT_SAT_ERR	0
ERR	Error while reading satellite configuration file	Wrong configuration file. No computation performed	XL_CFI_DEFAULT_SAT_INIT_READ_FILE_ERR	1

7.49xl_default_sat_close

7.49.1Overview

The **xl_default_sat_close** CFI function frees a default satellite id. that was initialized with **xl_default_sat_init,** so that it can be used again.

7.49.2Calling interface

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

```
#include <explorer_lib.h>
{
      long sat_id;
      xl_default_sat_close(&sat_id);
}
```

7.49.3Input parameters

The **xl_default_sat_close** function has the following input parameters:

Table 166: Input parameters of xl_default_sat_close function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
sat_id	long*	-	Satellite ID to free.	-	From XL_SAT_DEFAULT to XL_SAT_DEFAULT9

7.49.4Output parameters

This function does not return any value nor parameters.

7.49.5Warnings and errors

No warning nor errors are returned

7.50xl_set_tle_sat_data

7.50.1Overview

The **xl_set_tle_sat_data** CFI function changes the NORAD default SATCAT data associated to a given pre-defined satellite ID (the correspondence between satellite IDs and default SATCAT data is given in table 224 from the section 9.17 in [D_H_SUM]).
This function has to be called before reading and write files that are not compliant with such default values.

This function modifies static variables within the library itself, therefore it is not thread-safe. It is recommended to call this function before any other call to EOCFI functions and before any thread is started.

7.50.2Calling interface

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

```
#include <explorer_lib.h>
{
       long sat_id, status;
       long norad_sat_number;
       char norad_satcat[25];
       char int_des[9];
       status = xl_set_tle_sat_data (&sat_id,
                                     &norad_sat_number,
                                     norad_satcat,
                                     int_des);
}
```

7.50.3Input parameters

The **xl_set_tle_sat_data** function has the following input parameters:

Table 167: Input parameters of xl_set_tle_sat_data function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
sat_id	long*	-	Satellite ID	-	Any predefined satellite shown in Error: Reference source not found

7.50.4Output parameters

This function returns the status of the execution:

- 0 if the execution was correct

- -1 if an error occurred. This only could happen if the input sat_id was incorrect.

7.50.5Warnings and errors

No warning nor errors are returned

7.51xl_model_init

7.51.1 Overview

The **xl_model_init** CFI function initialises the model id with the requested models. There are two ways to initialise the model_id:

- Selecting a set of models via an enumeration value.
- Selecting a specific model for every model types.

Note that the model_id can be used if it has not been initilalised. In that case, the default CFI models are used.

The following table shows the possible models for every model type:

Table 168: Possible models for every model type

Model type	Models
Earth model (XL_MODEL_TYPE_EARTH)	XL_MODEL_EARTH_DEFAULT
Sun model (XL_MODEL_TYPE_SUN)	XL_MODEL_SUN_DEFAULT XL_MODEL_SUN_TRAVEL_TIME
Moon model (XL_MODEL_TYPE_MOON)	XL_MODEL_MOON_DEFAULT
Planet model (XL_MODEL_TYPE_PLANET)	XL_MODEL_PLANET_DEFAULT
Star model (XL_MODEL_TYPE_STAR)	XL_MODEL_STAR_DEFAULT
Nutation model (XL_MODEL_TYPE_NUTATION)	XL_MODEL_NUTATION_DEFAULT
Precession model (XL_MODEL_TYPE_PRECESSION)	XL_MODEL_PRECESSION_DEFAULT
Constants model (XL_MODEL_TYPE_CONSTANTS)	XL_MODEL_CONSTANTS_DEFAULT
Light propagation model (XL_MODEL_TYPE_LIGHT_PROPAGATION)	XL_MODEL_LIGHT_PROPAGATION_DISABLED XL_MODEL_LIGHT_PROPAGATION_RECEIVER XL_MODEL_LIGHT_PROPAGATION_TRANSMITTER

Table 169: Model sets

Model set	Selected Models
XL_MODEL_DEFAULT	XL_MODEL_EARTH_DEFAULT, XL_MODEL_SUN_DEFAULT, XL_MODEL_MOON_DEFAULT, XL_MODEL_PLANET_DEFAULT, XL_MODEL_STAR_DEFAULT, XL_MODEL_NUTATION_DEFAULT, XL_MODEL_PRECESSION_DEFAULT, XL_MODEL_CONSTANTS_DEFAULT, XL_MODEL_LIGHT_PROPAGATION_DISABLED
XL_MODEL_CONFIG	The models are chosen by the user with the models from Table 168

In order to simplify the initialisation, it is possible to select a set of models to be used.

Note 1: if XL_MODEL_SUN_TRAVEL_TIME is selected as Sun model, a compensation for the time needed for the light to travel from the Sun to the satellite is applied in the computations done by functions in the Pointing library related to Sun position with respect to the satellite.

Note 2: When the light propagation model is enabled, the target functions in the Pointing library keep into account the time spent by a generic signal travelling at the speed of light to go from the satellite to the target or vice versa.

7.51.2Calling interface

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

```
#include <explorer_lib.h>
{
      long mode, models[XL_NUM_MODEL_TYPES_ENUM];
      xl_model_id model_id = {NULL};
      long ierr[XL_NUM_ERR_MODEL_INIT], status;

      status =    xl_model_init (&mode, models,
                                 &model_id,
                                 ierr)
}
```

The `XL_NUM_MODEL_TYPES_ENUM and XL_NUM_ERR_MODEL_INIT` constant is defined in the file *explorer_lib.h.*

7.51.3Input parameters

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

Table 170: Input parameters of xl_model_init function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
mode	long	-	model set (according to Table 169)	-	-
models	long[]	all	These models are used in case of setting the *mode* parameter to XL_MODEL_CONFIG. The models are defined in Table 168	-	-
		0	Earth model	-	-
		1	Sun model		
		2	Moon model		
		3	Planet model		

	4	Star model		
	5	Nutation model		
	6	Precession model		
	7	Constants model		

7.51.4Output parameters

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

Table 171: Output parameters of xl_model_init function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_model_init	long	-	Status flag	-	-
model_id	xl_model_id	-	Model ID	-	-
ierr	long*	all	Error array	-	-

7.51.5Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_model_init** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 **xl_model_init** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 172: Error messages of xl_model_init function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Model ID is already initialised	No calculation performed	XL_CFI_MODEL_INIT_STATUS_ERR	0
ERR	Memory allocation error	No calculation performed	XL_CFI_MODEL_INIT_MEMORY_ERR	1
ERR	Wrong enumeration value for %s model	No calculation performed	XL_CFI_MODEL_INIT_WRONG_MODEL_ERR	2

7.52xl_model_close

7.52.1Overview

The **xl_model_close** CFI function cleans up any memory allocation performed by the initialization functions.

A complete calling sequence of the time reference computations is presented in section 4.2.

7.52.2Calling interface

The calling interface of the **xl_model_close** CFI function is the following:

```
#include <explorer_lib.h>
{
      xl_model_id model_id = {NULL};
      long ierr[XL_NUM_ERR_MODEL_CLOSE], status;
      status =    xl_model_close (&model_id, ierr);
}
```

7.52.3Input parameters

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

Table 173: Input parameters of xl_model_close function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
model_id	xl_model_id*	-	Structure that contains the time correlations.	-	-

7.52.4Output parameters

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

Table 174: Output parameters of xl_model_close function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_model_close	long	-	Status flag	-	-
ierr	long	-	Error vector	-	-

7.52.5Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_model_close** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 **xl_model_close** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 175: Error messages of xl_model_close function

Error type	Error message	Cause and impact	Error code	Error No
ERR	The Model Id is not initialized or it could be in use by another Id.	No calculation performed	XL_CFI_MODEL_CLOSE_WRONG_ID_ERR	0

7.53xl_model_get_data

7.53.1Overview

The **xl_model_get_data** CFI function returns a data structure contining the data used for the time initialisation.

7.53.2Calling interface

The calling interface of the **xl_model_get_data** CFI function is the following:

```
#include <explorer_lib.h>
{
      xl_model_id model_id;
      xl_model_id_data data;
      long status;
      status =    xl_model_get_data (&model_id, &data);
}
```

7.53.3Input parameters

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

Table 176: Input parameters of xl_model_get_data function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
model_id	xl_model_id*	-	Structure that contains the model information.	-	-

7.53.4Output parameters

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

Table 177: Output parameters of xl_model_get_data function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
xl_model_get_data	long	-	Status flag	-	-
data	xl_model_data	-	Model ID data	-	-

The data structure xl_model_data can be seen in Table 8.

7.53.5Warnings and errors

This function does not return any error/warning code. Only the status of the function indicates if the execution was correct or not.

The possible causes of error are:

- The model_id was not initialised.

7.54 xl_geoid_calc

7.54.1Overview

The **xl_geoid_calc** CFI function computes the geoid undulation, that is, the height of the geoid over the ellipsoid.
The geoid is computed at a given longitude and latitude according to the input model (default is EGM96 model). EGM96 is a geopotential model of the Earth consisting of spherical harmonic coefficients complete to degree and order 360. The nof_harmonics field in the input structure has to be set to the number of harmonics to be used in the computation (360 or less). The utc_time in the input structure is currently not used in the computation.

7.54.2Calling interface

The calling interface of the **xl_geoid_calc** CFI function is the following:

```
#include <explorer_lib.h>
{
      xl_geoid_calc_inputs geoid_calc_inputs;
      xl_geoid_calc_outputs geoid_calc_outputs;
      long ierr[XL_NUM_ERR_GEOID_CALC], status;
      status =    xl_geoid_calc (&geoid_calc_inputs,
                                 &geoid_calc_outputs,
                                 ierr);
}
```

7.54.3Input parameters

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

Table 178: Input parameters of xl_geoid_calc function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
geoid_calc_inputs	xl_geoid_calc_inputs*	-	Structure that contains the inputs needed for the computation.	-	-

7.54.4Output parameters

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

Table 179: Output parameters of xl_geoid_calc function

C name	C type	Array	Description	Unit	Allowed Range

		Element	(Reference)	(Format)	
geoid_calc_outputs	xl_geoid_calc_inputs*	-	Structure that contains the outputs of the computation,	-	-
ierr	long	-	Error vector	-	-

7.54.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_geoid_calc** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 **xl_geoid_calc** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 180: Error messages of xl_geoid_calc function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Wrong number of harmonics: %ld	The number of harmonics must be a number between 0 and 360. No calculation performed	XL_CFI_GEOID_CALC_WRONG_NUM_HARMONICS_ERR	0
ERR	Latitude (%lf) outside interval [-90.,90.]	No calculation performed	XL_CFI_GEOID_CALC_WRONG_LAT_ERR	1
ERR	Longitude (%lf) outside interval [0.,360.)	No calculation performed	XL_CFI_GEOID_CALC_WRONG_LON_ERR	2
ERR	Error computing the undulation from coefficients model	No calculation performed	XL_CFI_GEOID_CALC_UNDU_ERR	3

7.55 xl_quaternions_interpol

7.55.1Overview

The **xl_quaternions_interpol** CFI function performs, given 2 input quaternions, an interpolation for the requested time, obtaining the interpolated quaternion as output.

Notes:

1) the algorithm to be used for interpolation is given as input in xl_quaternions_interpol_cfg structure.
2) currently the supported algorithms are:
   - Slerp (see details: http://en.wikipedia.org/wiki/Slerp)
3) If the requested time is out of the interval defined by the input quaternions, then extrapolation is used and a warning is raised. The extrapolation degrades with the distance to the interval defined by input quaternions. In the following table the degradation for some time distances are shown:

Time out of interval [seconds]	Error in rotation angles [deg] (Quaternion time step 1 second)	Error in rotation angles [deg] (Quaternion time step 10 seconds)
1	0.000005	0.00005
10	0.0003	0.0005
100	0.06	0.07
500	0.96	0.95
1000	2.3	2.3

7.55.2Calling interface

The calling interface of the **xl_quaternions_interpol** CFI function is the following:

```
#include <explorer_lib.h>
{
      long ierr[XL_NUM_ERR_QUATERNIONS_INTERPOL], status;
      xl_quaternions_interpol_cfg quaternions_interpol_cfg;
      double q1[4], q2[4], q_out[4];
      double time_1, time_2, time_out;

       status = xl_quaternions_interpol(&quaternions_interpol_cfg, &time_1,
                q1, &time_2, q2, &time_out, q_out, ierr);
}
```

7.55.3Input parameters

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

Table 181: Input parameters of xl_quaternions_interpol

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
quaternions_interpol_cfg	xl_quaternions_interpol_cfg *	-	Structure that contains the inputs needed for the computation.	-	-
time_1	double *	-	Time for quaternion q1.	-	
q1	double *	-	First quaternion (for time_1). An array of size 4 must be passed.	-	
time_2	double *	-	Time for quaternion q2.	-	
q2	double *	-	Second quaternion (for time_2). An array of size 4 must be passed.	-	
time_out	double *	-	Time interpolation parameter.	-	t1 <= t <= t2

7.55.4Output parameters

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

Table 182: Output parameters of xl_quaternions_interpol

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
q_out	double *	-	Interpolated quaternion. An array of size 4 must be passed.	-	-
ierr	long	-	Error vector	-	-

7.55.5Warnings and errors

Next table lists the possible error messages that can be returned by the **xl_quaternions_interpol** CFI function after translating the returned extended status flag into the equivalent list of error messages by calling the function of the EO_LIB software library **xl_get_msg** (see [GEN_SUM]).

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 **xl_quaternions_interpol** function by calling the function of the EO_LIB software library **xl_get_code** (see [GEN_SUM]).

Table 183: Error messages of xl_quaternions_interpol function

Error type	Error message	Cause and impact	Error code	Error No
WARN	Time interpolation parameter out of quaternions range. Extrapolation will be used.	Calculation performed	XL_CFI_QUATERNIONS_INTERPOL_INPUT_TIME_OUT_WARN	0
ERR	Wrong algorithm. Only XL_INTERPOL_SLERP allowed.	No calculation performed	XL_CFI_QUATERNIONS_INTERPOL_WRONG_INPUT_ALGORITHM_ERR	1

8 CFI EXECUTABLE PROGRAMS

The following sections describe executables programs based on the CFI functions.

8.1 time_conv

This program makes time conversions between different formats and time references. It is call in the following way:

time_conv [**-ref_in**] input_time_ref
[**-ref_out**] output_time_ref
-fmt_in input_format
-fmt_out output_format
{**-day** j200_date (days)|
(**-t1** tranport_1 [**-t2** transport_2] [**-t3** transport_3] [**-t4** transport_4]) |
-date string_date (date)}
[**-v**]
[**-xd_v**]
[**-xl_v**]
[**-help**]
[**-show**]
[{ (**-tai** TAI_time **-gps** GPS_time **-utc** UTC_time **-ut1** UT1_time) |
(**-tmod** time_model **-tfile** time_file **-trid** time_reference
{(**-tm0** time0 **-tm1** time1) | (**-orb0** orbit0 **-orb1** orbit1) }) }]

Note that:

- Order of parameters does not matter.
- Bracketed parameters are not mandatory.
- Options between curly brackets and separated by a vertical bar are mutually exclusive.
- [**-xd_v**] option for EO_DATA_HANDLING Verbose mode.
- [**-xl_v**] option for EO_LIB Verbose mode.
- [**-v**] option for Verbose mode for all libraries (default is Silent).
- [**-show**] displays the inputs of the function and the results.
- Possible values for time_model: USER, NONE , IERS_B_PREDICTED, IERS_B_RESTITUTED, FOS_PREDICTED, FOS_RESTITUTED, DORIS_PRELIMINARY, DORIS_PRECISE, DORIS_NAVIGATOR.
- Possible values for time_ref and time_reference: TAI, UTC, UT1, GPS.

- Possible values for input_format and output_format:

  - Julian days: PROC

  - Transport format: TRANS_STD, TRANS_ENVI_GS, TRANS_CRYO_GS, TRANS_CRYO_TM, TRANS_CRYO_TM_SIRAL, SMOS_TM

  - date string: ASCII_STD, ASCII_STD_REF, ASCII_STD_MICROSEC, ASCII_STD_REF_MICROSEC, ASCII_COMPACT, ASCII_COMPACT_REF, ASCII_COMPACT_MICROSEC, ASCII_COMPACT_REF_MICROSEC, ASCII_ENVI, ASCII_ENVI_REF, ASCII_ENVI_MICROSEC, ASCII_ENVI_REF_MICROSEC, ASCII_CCSDSA, ASCII_CCSDSA_REF, ASCII_CCSDSA_MICROSEC, ASCII_CCSDSA_REF_MICROSEC, ASCII_CCSDSA_COMPACT, ASCII_CCSDSA_COMPACT_REF, ASCII_CCSDSA_COMPACT_MICROSEC, ASCII_CCSDSA_COMPACT_REF_MICROSEC

- The last three lines of parameters are used for initialising the time correlations . Note that only one set of parameters should be introduced

  - TAI, GPS, UTC and UT1 input times (as in xl_time_ref_init)

  - A file with time reference data, the time mode, the time reference name and a time range (as in xl_time_ref_init_file)

- In a time conversion, if the time refererence is not to be changed, the values for "-ref_in", "-ref_out" and the parameters for the time initialization are not needed. Note that the time reference will be always requested if the input/output format contains the reference in the date.

Examples:

```
time_conv -t1 1550 -t2 44266 -t3 176000 -t4 0 -fmt_in TRANS_STD
          -fmt_out PROC -ref_in TAI -ref_out TAI -v
          -tai 245.100000000000 -gps 245.099780092
          -utc 245.099594907407 -ut1 245.099587962


time_conv -date 2004-03-30T12:17:46.176000
          -fmt_in ASCII_CCSDSA_MICROSEC
          -fmt_out ASCII_STD -ref_in TAI -ref_out GPS -v
          -tai 245.100000000000 -gps 245.099780092
          -utc 245.099594907407 -ut1 245.099587962
```

9 RUNTIME PERFORMANCES

The library performance has been measured by dedicated test procedures run in 5 different platforms under the below specified machines:

OS ID	*Processor*	*OS*	*RAM*
LINUX64	Intel(R) Xeon(R) CPU E5-2470 0 @ 2.30GHz (16 cores)	GNU LINUX 2.6.24-16-generic (Ubuntu 8.04)	16 GB
LINUX32_LEGACY	Intel(R) Core(TM)2 Quad CPU Q8400 @ 2.66GHz	GNU LINUX 2.6.24-16-generic (Ubuntu 8.04)	4 GB
LINUX64_LEGACY	Intel(R) Core(TM)2 Quad CPU Q8400 @ 2.66GHz	GNU LINUX 2.6.24-16-generic (Ubuntu 8.04)	4 GB
MACIN64	Intel Core i7 4 cores @2,6 GHz	MAC OSX V10.10	16 GB
WINDOWS32	Intel(R) Xeon(R)CPU ES-2630 @ 2.40GHz 2.40GHz	Microsoft Windows 7	16 GB
WINDOWS64	Intel(R) Xeon(R)CPU ES-2630 @ 2.40GHz 2.40GHz	Microsoft Windows 7	16 GB

The table below shows the time (in miliseconds - ms) each function takes to be run under each platform:

Function ID	*WINDOWS 32*	*WINDOWS 64*	*LINUX64*	*LINUX64 LEGACY*	*LINUX32 LEGACY*	*MACIN 64*
xl_time_ref_init_file * File read: BULLETIN B 169. 874 records read. Depends on file reading implemented in explorer_data_handling.	4.620000	3.230000	1.200000	2.100000	2.700000	1.100000
xl_time_ref_init	0.001660	0.001145	0.000220	0.000270	0.000470	0.000480
xl_change_cart_cs	0.005599	0.004654	0.002350	0.003590	0.004530	0.001560
xl_cart_to_geod (mode = XL_CALC_POS_VEL)	0.004406	0.002827	0.001590	0.001530	0.003140	0.000940
xl_cart_to_geod (mode = XL_CALC_NO_ITER_POS)	0.002123	0.001308	0.000470	0.000460	0.000830	0.000400
xl_geod_to_cart	0.001791	0.001192	0.000370	0.000380	0.000680	0.000370
xl_cart_to_kepl	0.002402	0.001522	0.000650	0.000770	0.001130	0.000530
xl_kepl_to_cart	0.002267	0.001628	0.000690	0.000780	0.001280	0.000570
xl_geod_distance	0.009600	0.007800	0.005000	0.008000	0.008000	0.005000
xl_sun	0.013900	0.007800	0.010000	0.014000	0.020000	0.006000
xl_moon	0.015000	0.012900	0.140000	0.022000	0.022000	0.007000

Function ID	*WINDOWS 32*	*WINDOWS 64*	*LINUX64*	*LINUX64 LEGACY*	*LINUX32 LEGACY*	*MACIN 64*
xl_planet	0.015000	0.012500	0.012000	0.016000	0.019000	0.006000
xl_star_radec	0.012300	0.010300	0.009000	0.012000	0.014000	0.005000
xl_time_transport_to_ascii	0.002018	0.001803	0.001390	0.001900	0.002570	0.001240
xl_time_transport_to_transport	0.000203	0.000164	0.000120	0.000160	0.000240	0.000060
xl_time_transport_to_processing	0.000185	0.000154	0.000100	0.000140	0.000270	0.000070
xl_time_processing_to_ascii	0.002079	0.001852	0.001420	0.001900	0.002720	0.001250
xl_time_processing_to_transport	0.000213	0.000166	0.000120	0.000170	0.000360	0.000070
xl_time_processing_to_cuc	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
xl_time_cuc_to_processing	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
xl_time_processing_to_processing	0.000203	0.000157	0.000130	0.000170	0.000290	0.000070
xl_time_ascii_to_ascii	0.003083	0.002801	0.002290	0.003430	0.004720	0.002120
xl_time_ascii_to_transport	0.001315	0.001199	0.001010	0.001520	0.002170	0.000910
xl_time_ascii_to_processing	0.002619	0.002390	0.002010	0.003030	0.004350	0.001820
xl_time_get_leap_second_info	0.002661	0.002424	0.002020	0.003050	0.004390	0.001830
xl_time_init_status	0.000006	0.000006	0.000010	0.000000	0.000000	0.000010
xl_time_get_sat_id	0.000005	0.000004	0.000003	0.000004	0.000007	0.000003
xl_time_get_mode	0.000005	0.000004	0.000003	0.000003	0.000007	0.000003
xl_time_get_id	0.000178	0.000242	0.000060	0.000090	0.000180	0.000110
xl_time_set_id	0.000184	0.000229	0.000090	0.000100	0.000140	0.000110
xl_time_add	0.000034	0.000025	0.000010	0.000020	0.000040	0.000010
xl_time_diff	0.000043	0.000031	0.000030	0.000030	0.000040	0.000020
xl_time_obt_to_time	0.000124	0.000057	0.000040	0.000050	0.000140	0.000030
xl_time_time_to_obt	0.000125	0.000064	0.000040	0.000060	0.000150	0.000030
xl_get_rotation_angles	0.000948	0.000850	0.000350	0.000420	0.000670	0.000230
xl_get_rotated_vectors	0.000690	0.000609	0.000310	0.000820	0.001010	0.000200
xl_position_on_orbit	0.005760	0.004861	0.002770	0.004210	0.004930	0.001730
xl_cart_to_radec	0.001998	0.001324	0.000650	0.000870	0.001160	0.000430
xl_radec_to_cart	0.001975	0.001298	0.000620	0.000720	0.001220	0.000410
xl_euler_to_matrix	0.000258	0.000155	0.000190	0.000650	0.000390	0.000110
xl_matrix_to_euler	0.000573	0.000358	0.000190	0.000240	0.000340	0.000090
xl_star_catalog	0.031344	0.025677	0.022310	0.028510	0.036560	0.010780
xl_topocentric_to_ef	0.005310	0.003900	0.002600	0.003300	0.004000	0.002000
xl_ef_to_topocentric	0.005270	0.003810	0.002800	0.003300	0.004400	0.001500
xl_vectors_to_quaternions	0.000310	0.000300	0.000100	0.000100	0.000100	0.000000
xl_quaternions_to_vectors	0.000080	0.000080	0.000000	0.000100	0.000100	0.000000

Function ID	*WINDOWS 32*	*WINDOWS 64*	*LINUX64*	*LINUX64 LEGACY*	*LINUX32 LEGACY*	*MACIN 64*
xl_default_sat_init * called with xl_default_sat_close	0.000080	0.000060	0.000100	0.000100	0.000100	0.000100
xl_quaternions_interpol	0.000210	0.000150	0.000100	0.000100	0.000400	0.000100

Note that when the value “0.000000” is defined for a function in a certain platform, it means that its running time is lower than 1 nano-second and so it can be considered as “0”.

10 LIBRARY PRECAUTIONS

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

- When a message like:

  <LIBRARY NAME> >>> ERROR in *xl_function*: Internal computation error # n

  or

  <LIBRARY NAME> >>> WARNING in *xl_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.