Earth Explorer
Mission CFI Software

EXPLORER_GEN_FILES
SOFTWARE USER MANUAL

Code:	CS-MA-DMS-GS-0007
Issue:	3.0.1
Date:	13/08/04

	Name	**Function**	**Signature**
Prepared by:	José Antonio González Abeytua	Project Manager	
	Andrés Canales Molina	Project Engineer	
	Fco. Jesús López Honrubia	Project Engineer	
	Juan José Borrego Bote	Project Engineer	
Checked by:	José Antonio González Abeytua	Project Manager	
Approved by:	José Antonio González Abeytua	Project Manager	

DEIMOS Space S.L.
Ronda de Poniente, 19,
Edificio Fiteni VI, Portal 2, 2ª Planta, Tres Cantos
28760 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:	Framemaker 6.0
Archive Code:	P/SUM/DMS/01/026-026
Electronic file name:	cs-ma-dms-gs-0007-21

Document Status Log

Issue	Change Description	Date	Approval
1.0	New document	23/07/02	
2.0	Maintenance release with the following main changes: • general overview of library added. • updated interfaces to create complete filenames and fixed headers • updated error handling See change bars for complete document update.	29/11/02	
2.1	Maintenance release with the following main changes: • Executable programs added. • Function to generate **DORIS Navigator** file. See change bars for complete document update.	13/05/03	
2.2	Maintenance release. See change bars for complete document update.	30/09/03	
3.0	New initialisation strategy and interfaces.	21/07/04	

Table of Contents

List of Tables

List of Figures

1 SCOPE

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

2 ACRONYMS AND NOMENCLATURE

2.1 Acronyms

ANX	Ascending Node Crossing
AOCS	Attitude and Orbit Control Subsystem
CFI	Customer Furnished Item
EF	Earth Fixed reference frame
ESA	European Space Agency
ESTEC	European Space Technology and Research Centre
FOS	Flight Operations Segment
GS	Ground Station
OBT	On-board Binary Time
OSF	Orbit Scenario File
POF	Predicted Orbit File
ROF	Restituted Orbit File
SSP	Sub-Satellite Point
SRAR	Satellite Relative Actual Reference
SUM	Software User Manual
TOD	True of Date reference frame
UTC	Universal Time Coordinated
UT1	Universal Time UT1
WGS[84]	World Geodetic System 1984

2.2 Nomenclature

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

3 APPLICABLE AND REFERENCE DOCUMENTS

3.1 Applicable documents

[GEN_SUM] Earth Explorer Mission CFI Software. General Software User Manual. CS-MA-DMS-GS-0002. Issue 3.0. 21/07/04

3.2 Reference documents

[MCD] Earth Explorer Mission CFI Software. Mission Conventions Document. CS-MA-DMS-GS-0001. Issue 1.4 21/07/04.

[F_H_SUM] Earth Explorer Mission CFI Software. EXPLORER_FILE_HANDLING Software User Manual. CS-MA-DMS-GS-0008. Issue 3.0. 21/07/04

[LIB_SUM] Earth Explorer Mission CFI Software. EXPLORER_LIB Software User Manual. CS-MA-DMS-GS-0003. Issue 3.0. 21/07/04

[ORBIT_SUM] Earth Explorer Mission CFI Software. EXPLORER_ORBIT Software User Manual. CS-MA-DMS-GS-0004. Issue 3.0. 21/07/04

[POINT_SUM] Earth Explorer Mission CFI Software. EXPLORER_POINTING Software User Manual. CS-MA-DMS-GS-0005. Issue 3.0. 21/07/04

[VISIB_SUM] Earth Explorer Mission CFI Software. EXPLORER_VISIBILITY Software User Manual. CS-MA-DMS-GS-0005. Issue 3.0. 21/07/04

[FORMATS] Earth Explorer File Format Guidelines. CS-TN-ESA-GS-0148.

4 INTRODUCTION

4.1 Functions Overview

This software package contains the CFI functions required to compute the orbit scenario file, used for Earth Explorer mission planning purposes, the instrument swath template files, used for instance by the time segment calculation CFI functions of EXPLORER_VISIBILITY, the Instrument Mode Transition File, used in the planning of CRYOSAT, and several orbit files useful for testing purposes (Predicted Orbit File, Restituted Orbit File, DORIS Navigator Files).

It contains:

- a library of functions which can be called from a main executable program
- a set of executable programs (1 for each function) with the exact same functionality as the functions

This library is to be used for file generation to facilitate the planning and/or testing of Earth Explorer satellite operations.

The EXPLORER_GEN_FILES library includes the following CFI functions:

- **xg_gen_osf_create**: generates the orbit scenario file with user provided inputs
- **xg_gen_osf_append_orbit_change**: adds an orbit change to a previously generated OSF
- **xg_gen_osf_change_repeat_cycle**: adds an orbit change for a given target orbit to an existing OSF.
- **xg_gen_osf_add_drift_cycle**: adds an orbit change for a requested orbit with a particular ascending node longitude and an orbit for the manoeuvre.
- **xg_gen_swath** generates the instrument swath template file for a given satellite, instrument mode and orbit.
- **xg_gen_pof**: generates a Predicted Orbit File from several different reference input files.
- **xg_gen_rof** and **xg_gen_rof_prototype**: generates a Restituted Orbit File from several different reference input files.
- **xg_gen_dnf**: generates a DORIS Navigator File from several different reference input files.
- **xg_gen_imf**: generate an Instrument Mode Transition File containing instrument mode change events.

Several files are required to operate properly the above functions:

- Orbit Scenario File (**all functions**)
- Instrument Swath Definition Files (**xg_gen_swath**)
- Predicted, Reference and DORIS files (**xg_gen_rof, xg_gen_pof**, **xg_gen_dnf**)

4.2 Calling Sequence

An overview of the EXPLORER_GEN_FILES data flow is presented in the following figure:

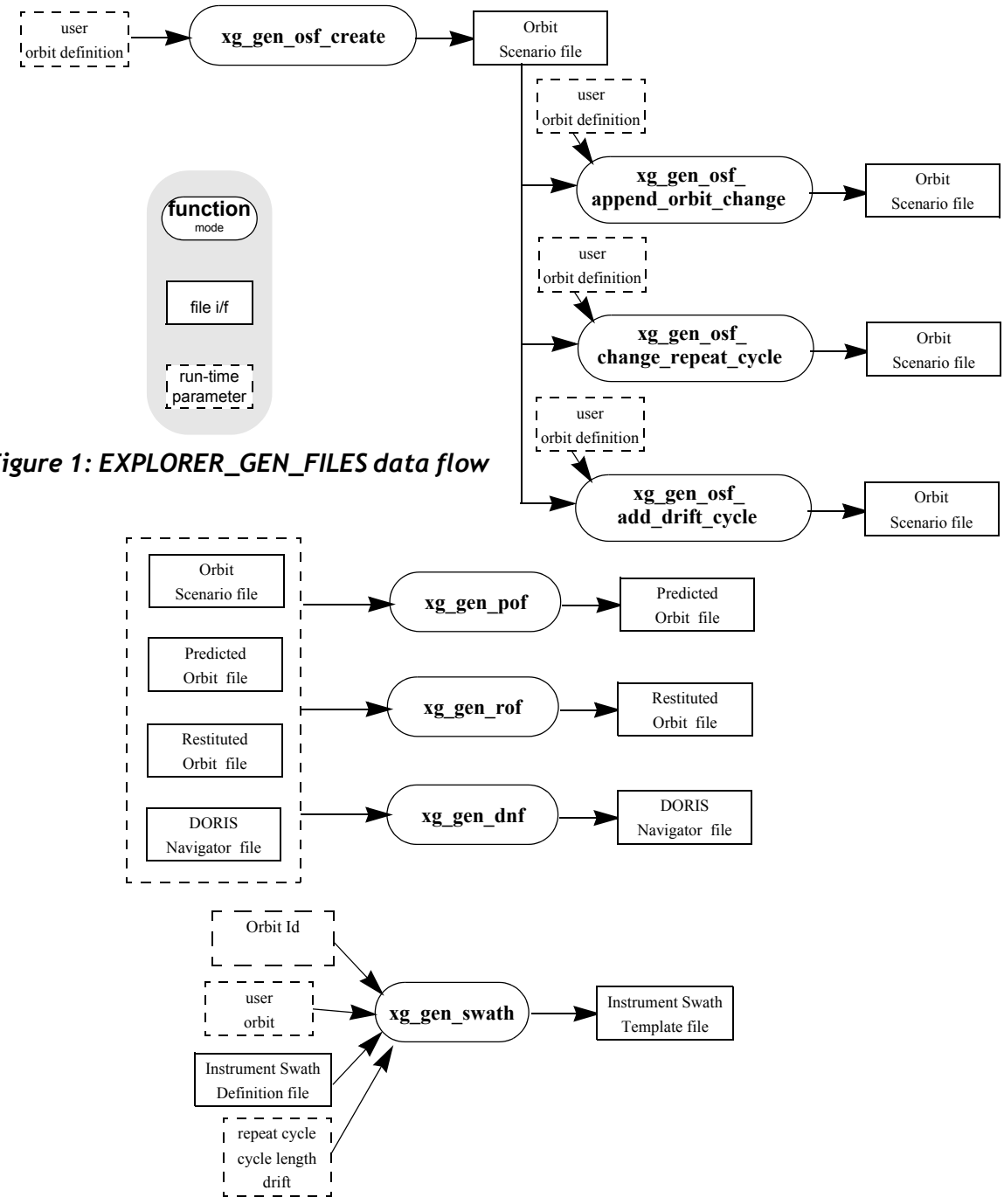

Figure 1: EXPLORER_GEN_FILES data flow

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, reference frames, parameters and models used in this document.
- [GEN_SUM] for a complete overview of the CFI, and in particular the detailed description of *Id* concept and usage and error handling functions.

5 LIBRARY INSTALLATION

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

Note that example data files are provided with this CFI:

- Orbit files to be used with ***xg_gen_pof,*** and ***xg_gen_rof***
- Orbit scenario files to be used with ***xg_gen_swath, xg_gen_pof*** and ***xg_gen_rof***
- Swath definition files to be used with ***xg_gen_swath***

These files are orbit file examples.

6 LIBRARY USAGE

Note that to use the EXPLORER_GEN_FILES software library, the following other CFI software libraries are required:

- EXPLORER_FILE_HANDLING version 3.0 (See [F_H_SUM]).
- EXPLORER_LIB version 3.0 (See [F_H_SUM]).
- EXPLORER_ORBIT version 3.0 (See [ORBIT_SUM]).
- EXPLORER_POINTING version 3.0 (See [POINT_SUM])

It is also needed to have properly installed in the system the following external GPL libary:

- LIBXML2 (see [GEN_SUM]).

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

- explorer_gen_files.h (for a C application)
- explorer_gen_files.inc (for a ForTran application under SOLARIS/Linux)
- explorer_gen_files_win.inc (for a ForTran application under Windows 95/98/NT/2000)

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

- Solaris/Linux:

  -I*cfi_include_dir* -L*cfi_lib_dir* -lexplorer_gen_files -lexplorer_pointing -lexplorer_orbit -lexplorer_lib -lexplorer_file_handling -lxml2

- Windows users:

  /I *"cfi_include_dir"* /libpath:*"cfi_lib_dir"* libexplorer_gen_files.lib libexplorer_pointing.lib libexplorer_orbit.lib libexplorer_lib.lib libxml2.lib

- MacOS:

  -I*cfi_include_dir* -L*cfi_lib_dir* -lexplorer_gen_files -lexplorer_pointing -lexplorer_orbit -lexplorer_lib -lexplorer_file_handling -framework libxml -framework libiconv

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

To avoid problems in linking a user application with the EXPLORER_GEN_FILES 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 XG_ or xg_.

This is summarized in table 1.

Table 1: CFI functions included within EXPLORER_GEN_FILES library

Function Name	Enumeration value	long
Main CFI Functions		
xg_gen_swath	XG_GENSWATH_ID	1
xg_gen_osf_create	XG_GEN_OSF_CREATE_ID	2
xg_gen_osf_append_orbit_change	XG_GEN_OSF_APPEND_ORBIT_CHANGE_ID	3
xg_gen_osf_change_repeat_cycle	XG_GEN_OSF_CHANGE_REPEAT_CYCLE_ID	4
xg_gen_osf_add_drift_cycle	XG_GEN_OSF_ADD_DRIFT_CYCLE_ID	5
xg_gen_rof	XG_GEN_ROF_ID	6
xg_gen_rof_prototype_id	XG_GEN_ROF_PROTOTYPE_ID	7
xg_gen_pof	XG_GEN_POF_ID	8
xg_gen_dnf	XG_GEN_DNF_ID	9
Error Handling Functions		
xg_verbose	not applicable	
xg_silent		
xg_get_code		
xg_get_msg		
xg_print_msg		

Notes about the table:

- To transform the status vector returned by a CFI function to either a list of error codes or list of error messages, the enumeration value (or the corresponding integer value) described in the table must be used.
- The error handling functions have no enumerated value.

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 (XG_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 EXPLORER_GEN_FILES routines, as shown in the table below. The enumerations presented in [GEN_SUM] are also applicable.

Table 2: Some enumerations within EXPLORER_GEN_FILES library

Input	Description	Enumeration value	Long
Phase increment	Do not increment phase number at next orbit change	XG_NO_PHASE_INCREMENT	0
	Do increment phase number at next orbit change	XG_PHASE_INCREMENT	1
Orbit change search direction	Search forward	XG_SEARCH_FORWARD	1
	Search backward	XG_SEARCH_BACKWARD	-1
File Type	Orbit Scenario File	XG_REF_FILETYPE_OSF	1
	FOS Predicted Orbit File	XG_REF_FILETYPE_POF	2
	DORIS Navigator File	XG_REF_FILETYPE_DORIS_NAV	3
	FOS Restituted Orbit File	XG_REF_FILETYPE_ROF	4
	DORIS Preliminary Orbit File	XG_REF_FILETYPE_DORIS_PREM	5
	DORIS Precise Orbit File	XG_REF_FILETYPE_DORIS_PREC	6
Precission for ROF and DORIS state vectors times	Default value, non-precise	XG_OSV_PRECISE_NO	1
	Precise location every integer minute	XG_OSV_PRECISE_MINUTE	2
	Precise location every ten seconds	XG_OSV_PRECISE_TEN_SECONDS	3

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**.

7 CFI FUNCTIONS DESCRIPTION

The following sections describe each CFI function.

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

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

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

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

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

7.1 xg_gen_swath

7.1.1 Overview

The **xg_gen_swath** function generates for the different instrument modes the corresponding instrument swath template file. These template files define the swaths to be used in the segment calculation routines of **explorer_visibility**.

The **xg_gen_swath** function contains for each instrument swath type a swath calculation algorithm. The selection of the algorithm depends on the parameters of the corresponding swath definition found in the instrument swath definition file (one instrument swath definition file can contain several instrument swath definitions). Remark that for ENVISAT-1 files, the algorithm is explicitly specified in the swath definition, but for swath definition files in XML format the algorithm is deduced from the type of swath, the geometry and other instrument dependent parameters. There is an example of a swath definition file in the Appendix A.

The instrument swath template file, consists of a header (TBC for files in XML format) which contains the altitude range of the swath. The data block contains for *n* (between 50 and 6000, typically 1200) equally spread times along one orbit, the location of the swath, for 1(for point swath types) or 3 points. These points are located from left to right when looking in the flight direction (e.g. for ENVISAT ASAR: from near-swath, via mid-swath, to far-swath). For a drawing describing the swath configuration, please refer to [VISIB_SUM].

For Earth-fixed swaths, the location is given in longitude and latitude, in degrees, for the orbit with a longitude of ascending node of 0.0 degrees. For Inertial swaths, the location is the direction in inertial space (True of Date) in Right Ascension and Declination, in degrees, for the orbit with a Right Ascension of Ascending Node of 0.0 degrees.

The instrument swath template files are only dependent on:

- The instrument swath definition file
- The repeat cycle and cycle length

7.1.2 Calling interface

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

```
#include <explorer_gen_files.h>
{
      xo_orbit_id orbit_id;
      xp_atmos_id atmos_id;
      long requested_orbit,
           repeat_cycle, cycle_length, drift_mode, version_number;
      double mlst_drift, inclination;
      char *swath_definition_file;
      char *instr_swath_file_suffix, *dir_name, *file_class,
           *fh_system;
      long status, ierr[XG_ERR_VECTOR_MAX_LENGTH];
      status = xg_gen_swath (&orbit_id, &atmos_id,
                             &requested_orbit, swath_definition_file,
```

```
                              &repeat_cycle, &cycle_length,
                              &drift_mode,
                              &mlst_drift, &inclination,
                              dir_name, instr_swath_file_suffix,
                              file_class, &version_number, fh_system,
                              ierr);

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

        status = xg_gen_swath_run (&run_id,
                              &requested_orbit, swath_definition_file,
                              &repeat_cycle, &cycle_length,
                              &drift_mode,
                              &mlst_drift, &inclination,
                              dir_name, instr_swath_file_suffix,
                              file_class, &version_number, fh_system,
                              ierr);
}
```

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

```
#include <explorer_gen_files.inc>

        INTEGER*4 SAT_ID, REQUESTED_ORBIT
        INTEGER*4 REPEAT_CYCLE, CYCLE_LENGTH, DRIFT_MODE, VERSION_NUMBER
        REAL*8 MLST_DRIFT, INCLINATION
        CHAR*(*) ORBIT_SCENARIO_FILE, SWATH_DEFINITION_FILE
        CHAR*(*) INSTR_SWATH_FILE_SUFFIX, DIR_NAME, FILE_CLASS, FH_SYSTEM
        INTEGER*4 STATUS, IERR(XG_ERR_VECTOR_MAX_LENGTH)

        STATUS = XG_GEN_SWATH( SAT_ID, ORBIT_SCENARIO_FILE,
      &                        REQUESTED_ORBIT, SWATH_DEFINITION_FILE,
      &                        REPEAT_CYCLE, CYCLE_LENGTH,
      &                        DRIFT_MODE,
      &                        MLST_DRIFT, INCLINATION,
      &                        DIR_NAME, INSTR_SWATH_FILE_SUFFIX,
      &                        FILE_CLASS, VERSION_NUMBER, FH_SYSTEM,
      &                        IERR)
```

7.1.3 Input parameters

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

Table 3: Input parameters of xg_gen_swath function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
orbit_id	xo_orbit_id*	-	Structure that contains the orbit data.	-	-
atmos_id	xp_atmos_id*	-	Structure that contains the atmosphere initialisation. (Needed only if the swath definition file requires atmosphere initialisation).	-	-
requested_orbit	long*	-	Orbit for which the instrument swath template file will be calculated. If non zero, then repeat_cycle, cycle_length and mlst_drift of the orbit are looked up in the orbit_scenario_file If set to zero, then use directly repeat_cycle, cycle_length and mlst_drift parameters (see below)	absolute orbit number	= 0 or >= start_osf
swath_definition_file	char*	-	File name of the instrument swath definition file	-	-
repeat_cycle	long*	-	Repeat cycle of the reference orbit (only used if requested_orbit is zero)	days	>= 1
cycle_length	long*	-	Cycle length of the reference orbit (only used if requested_orbit is zero)	orbits	>= 14
drift_mode	long*	-	Flag to select between drift in mean local solar time and inclination as input characterization of the reference orbit (only used if requested_orbit is zero)	-	[0,1]
mlst_drift	double*	-	If *drift_mode = XO_NOSUNSYNC_DRIFT* Drift in mean local solar time of the reference orbit: · MLST[N+1]=MLST[N]+MLST-drift	seconds/day	TBD
inclination	double*	-	If *drift_mode = XO_NOSUNSYNC_INCLINATION* Inclination of the reference orbit	deg	[0,180]

Table 3: Input parameters of xg_gen_swath function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
dir_name	char*	-	Directory where the resulting STF is written (if empty (i.e. ""), the current directory is used)	-	-
instr_swath_file_suffix	char*	-	name suffix for output swath file if empty (i.e. ""), the software will generate the suffix according to file name specification pre-sented in [FORMATS]	-	-
file_class	char*	-	File class for output swath file	-	-
version_number	long*	-	Version number of output swath file	-	>= 0
fh_system	char*	-	System field of the output swath file fixed header	-	-

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

- Drift mode: mlst_drift. See [ORBIT_SUM].

7.1.4 Output parameters

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

Table 4: Output parameters of xg_gen_swath function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
ierr[XG_ERR_VECTOR_MAX_LENGTH]	long	all	Status vector	-	-

7.1.5 Warnings and errors

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

Table 5: Error messages of xg_gen_swath function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Cannot open Swath Definition File.	Computation not perfomed	XG_CFI_GENSWATH_OPEN_SDF_ERR	0
ERR	Input Orbit ID is not initialized	Computation not perfomed	XG_CFI_GENSWATH_ORBIT_INIT_ERR	1
ERR	Unable to store Swath Template File names (not enough memory).	Computation not perfomed	XG_CFI_GENSWATH_STF_FILENAMES_ERR	2
ERR	Error in \"SWATH=\". There is no cycle length value. Swath Definition File record: %ld.	Computation not perfomed	XG_CFI_GENSWATH_SWATH_ID_CYCL_LENGTH_1ST_ERR	3
ERR	Cycle length value in \"SWATH=\" does not match with cycle length for computations. Swath Definition File record: %ld	Computation not perfomed	XG_CFI_GENSWATH_SWATH_ID_CYCL_LENGTH_2ND_ERR	4
WARN	Cannot open Swath Template File for writting because the input directory does not exist. Swath Template File: %ld to be written in the current directory.	Computation perfomed. Message to inform the user.	XG_CFI_GENSWATH_NO_DIR_WARN	5
ERR	Cannot open Swath Template File for writting. Swath Template File: %ld failed	Computation not perfomed	XG_CFI_GENSWATH_OPEN_STF_WRITE_ERR	6
ERR	Error while writing beginning of Swath Template File. Swath Template File: %ld failed.	Computation not perfomed	XG_CFI_GENSWATH_STF_BEG_WRITE_ERR	7
ERR	Error while writing comment line. Swath Template File: %ld failed.	Computation not perfomed	XG_CFI_GENSWATH_COMM_LINE_ERR	8
ERR	Cannot rename Swath Template File: %ld.	Computation not perfomed	XG_CFI_GENSWATH_RENAME_FILES_ERR	9

Table 5: Error messages of xg_gen_swath function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Cannot create XML tree.	Computation not perfomed	XG_CFI_GENSWATH_CREATE_XML_ERR	10
ERR	Cannot create root node in the XML tree.	Computation not perfomed	XG_CFI_GENSWATH_CREATE_ROOT_XML_ERR	11
ERR	Cannot create a node in the XML tree.	Computation not perfomed	XG_CFI_GENSWATH_CREATE_NODE_XML_ERR	12
ERR	Cannot set a value to a node in the XML tree.	Computation not perfomed	XG_CFI_GENSWATH_SET_VALUE_XML_ERR	13
ERR	Cannot write to disk the XML tree.	Computation not perfomed	XG_CFI_GENSWATH_WRITE_XML_ERR	14
ERR	Cannot navigate through the XML tree.	Computation not perfomed	XG_CFI_GENSWATH_NAV_XML_ERR	32
ERR	Error in Genswath inputs (REPEAT CYCLE, CYCLE LENGTH or Orbit Scenario File inconsistency).	Computation not perfomed	XG_CFI_GENSWATH_XG_CHECK_GENSW_INP_ERR	33
ERR	Error while generating Ascending Node parameters.	Computation not perfomed	XG_CFI_GENSWATH_XG_GENSTATE_ERR	34
ERR	Error in Genswath inputs.	Computation not perfomed	XG_CFI_GENSWATH_XG_SDF_PREV_CHECK_ERR	35
ERR	“SWATH=“ value of Swath record: %ld is repeated below.	Computation not perfomed	XG_CFI_GENSWATH_PG_SDF_PREV_CHECK_SWATH_VALUE_ERR	36
ERR	Error reading Swath record: %ld.	Computation not perfomed	XG_CFI_GENSWATH_PG_SDF_PREV_CHECK_SWATH_RECORD_ERR	37
ERR	Error reading Swath Definition File header.	Computation not perfomed	XG_CFI_GENSWATH_PG_SDF_READ_HEADER_ERR	38
ERR	Error reading Swath Definition File record: %ld.	Computation not perfomed	XG_CFI_GENSWATH_PG_SDF_REC_READ_ERR	39
ERR	Error in algorithm computations. Swath Template File: %ld failed.	Computation not perfomed	XG_CFI_GENSWATH_PG_ALGOR_ERR	40
ERR	Error while generating output filename. Swath Template File: %ld failed.	Computation not perfomed	XG_CFI_GENSWATH_PG_FILENAME_GENER_SWATH_ERR	41
ERR	Error writing fixed header. Swath Template File: %ld failed.	Computation not perfomed	XG_CFI_GENSWATH_PG_FHR_WRITE_ERR	42

Table 5: Error messages of xg_gen_swath function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Error while writing Swath Template File variable header. Swath Template File: %ld failed.	Computation not perfomed	XG_CFI_GENSWATH_PG_STF_VHR_WRITE_ERR	43
ERR	Satellite ID not recognised.	Computation not perfomed	XG_CFI_GENSWATH_SAT_ERR	44

7.1.6 Runtime performances

The following runtime performance has been measured.

Table 6: Runtime performances of xg_gen_swath function

Ultra Sparc [ms]
TBD

7.1.7 Executable Program

The **gen_swath** executable program can be called from a Unix shell as:

gen_swath **-sat** satellite_name
-sdf swath_definition_file_name
{(-osf orbit_scenario_file_name **-orbit** orbit_number)|
(**-repcyc** repeat_cycle **-cyclen** cycle_length {**-mlstdr** mlst_drift | **-inc** inclination})**}**
[**-dir** dir_name] (current directory by default)
[**-isfx** instrument_swath_file_suffix] (empty string by default)
[**-flcl** file_class] (empty string by default)
[**-vers** version] (version=0 by default)
[**-fhsys** fh_system] (empty string by default)
[**-v**]
[**-xl_v**]
[**-xo_v**]
[**-xp_v**]
[**-xg_v**]
[**-help**]
[**-show**]
{(-tai TAI_time **-gps** GPS_time **-utc** UTC_time **-ut1** UT1_time) |
(-tmod** time_model **-tfile** time_reference_data file **-trid** time_reference
{(**-tm0** time 0 **-tm1** time 1) | (**-orb0** orbit 0 **-orb1** orbit 1) })**}**

Note that:

- Order of parameters does not matter.
- Bracketed parameters are not mandatory (For example, if **-isfx** argument is not provided, instrument_swath_file_name_suffix is considered to be an empty string).
- Options between curly brackets and separated by a vertical bar are mutually exclusive (For example, that lines 3 and 4 are mutually exclusive).
- [**-xl_v**] option for EXPLORER_LIB Verbose mode.
- [**-xo_v**] option for EXPLORER_ORBIT Verbose mode.
- [**-xp_v**] option for EXPLORER_POINTING Verbose mode.
- [**-xg_v**] option for EXPLORER_GEN_FILES 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 *satellite_name*: ERS1, ERS2, ENVISAT, METOP1, METOP2, METOP3, CRYOSAT, ADM, GOCE, SMOS.

Example:

```
gen_swath  -sat ENVISAT -orbit 2000 -osf ACCEPTANCE_OSF.N1
           -sdf SDF_MERIS.1200pts.N1 -xg_v -isfx gen_swath_at_301
           -dir ./gen_swath
```

7.2 xg_gen_osf_create

7.2.1 Overview

The **xg_gen_osf_create** CFI function creates a reference Orbit Scenario File (OSF) with one orbit change data structure using only user inputs in the calling interface. This data structure characterizes the reference orbit by means of the following parameters:

- Absolute orbit number
- Relative orbit number
- Cycle number
- Phase number
- Repeat cycle (days)
- Cycle length (orbits)
- Ascending crossing node longitude
- Mean local solar time of the ascending crossing node
- Mean local solar time drift (seconds per day)
- Time of the ascending crossing node (TAI, UTC and UT1)

7.2.2 Calling interface

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

```
#include <explorer_gen_files.h>
{
      long sat_id;
      xl_time_id time_id;
      long abs_orbit_number, cycle_number, phase_number,
           repeat_cycle, cycle_length, drift_mode, version_number;
      double anx_long, inclination, mlst_drift, mlst, date;
      char output_dir[XG_MAX_LENGTH], output_filename[XG_MAX_LENGTH];
      char *file_class, *fh_system;
      long status, ierr[XG_ERR_VECTOR_MAX_LENGTH];
      status = xg_gen_osf_create (&sat_id, &time_id, &abs_orbit_number,
                                  &cycle_number, &phase_number,
                                  &repeat_cycle, &cycle_length,
                                  &anx_long, &drift_mode,
                                  &inclination, &mlst_drift,
                                  &mlst, &date,
                                  output_dir, output_filename,
                                  file_class, &version_number,
                                  fh_system,
                                  ierr);
```

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

        status = xg_gen_osf_create_run (&run_id, &abs_orbit_number,
                                        &cycle_number, &phase_number,
                                        &repeat_cycle, &cycle_length,
                                        &anx_long, &drift_mode,
                                        &inclination, &mlst_drift,
                                        &mlst, &date,
                                        output_dir, output_filename,
                                        file_class, &version_number,
                                        fh_system,
                                        ierr);
}
```

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

```
#include <explorer_gen_files.inc>

        INTEGER*4 SAT_ID, ABS_ORBIT_NUMBER, CYCLE_NUMBER, PHASE_NUMBER
        INTEGER*4 REPEAT_CYCLE, CYCLE_LENGTH, DRIFT_MODE, VERSION_NUMBER
        REAL*8 ANX_LONG, INCLINATION, MLST_DRIFT, MLST, DATE
        CHAR*1 OUTPUT_DIR(XG_MAX_LENGTH), OUTPUT_FILENAME(XG_MAX_LENGTH)
        CHAR*1 FILE_CLASS, FH_SYSTEM
        INTEGER*4 STATUS, IERR(XG_ERR_VECTOR_MAX_LENGTH)

        STATUS = XG_GEN_OSF_CREATE ( SAT_ID, ABS_ORBIT_NUMBER,
      &                              CYCLE_NUMBER, PHASE_NUMBER,
      &                              REPEAT_CYCLE, CYCLE_LENGTH,
      &                              ANX_LONG, DRIFT_MODE,
      &                              INCLINATION, MLST_DRIFT,
      &                              MLST, DATE,
      &                              OUTPUT_DIR, OUTPUT_FILENAME,
      &                              FILE_CLASS, VERSION_NUMBER,
      &                              FH_SYSTEM,
      &                              IERR)
```

7.2.3 Input parameters

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

Table 7: Input parameters of xg_gen_osf_create 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.	-	-
abs_orbit_number	long*	-	Orbit number in OSF first orbit change	-	>= 1
cycle_number	long*	-	Cycle number in OSF first orbit change	-	>= 1
phase_number	long*	-	Phase number in OSF first orbit change	-	>= 1
repeat_cycle	long*	-	Repeat cycle of the reference orbit	days	>= 1
cycle_length	long*	-	Cycle length of the reference orbit	orbits	>= 14
anx_long	double*	-	Reference orbit ascending node crossing longitude	deg	[-180, 180]
drift_mode	long*	-	Flag to select between drift in mean local solar time and inclination as input characterization of the reference orbit	-	[0,1]
inclination	double*	-	If *drift_mode* = *XO_NOSUNSYNC_INCLINATION* Inclination of the reference orbit	deg	[0,180]
mlst_drift	double*	-	If *drift_mode* = *XO_NOSUNSYNC_DRIFT* Drift in mean local solar time of the reference orbit: · MLST[N+1]=MLST[N]+MLST-drift	seconds/day	TBD
mlst	double*	-	Mean local solar time at ascending node	decimal hours	[0,24)
date	double*	-	ANX date	decimal days	-
output_dir	char*	-	Directory where the resulting OSF is written (if empty (i.e. ““), the current directory is used)	-	-
output_filename	char*	-	Output OSF name if empty (i.e. ““), the software will generate the filename according to file name specification presented in [FORMATS]	-	-

Table 7: Input parameters of xg_gen_osf_create function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
file_class	char*	-	File class for output Orbit file	-	-
version_number	long*	-	Version number of output Orbit file	-	>= 0
fh_system	char*	-	System field of the output Orbit file fixed header	-	-

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

- Satellite ID: sat_id. See [GEN_SUM].
- Drift mode: mlst_drift. See [ORBIT_SUM].

This CFI can generate Orbit Scenario Files for both sun-synchronous orbits and quasi-sun-synchronous orbits.

Use `drift_mode=XO_NOSUNSYNC_DRIFT` and `mlst_drift = 0.0` for a sun-synchronous orbit.

Use any other combination for the general case of quasi-sun-synchronous orbit.

7.2.4 Output parameters

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

Table 8: Output parameters of xg_gen_osf_create function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
ierr[XG_ERR_VECTOR_MAX_LENGTH]	long	all	Status vector	-	-

7.2.5 Warnings and errors

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

Table 9: Error messages of xg_gen_osf_create function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Wrong input values	Wrong value of one or more of the following input parameters: abs_orbit_number, cycle_number, phase_number, repeat_cycle, cycle_length, mlst Computation not performed	XG_CFI_GEN_OSF_CREATE_INPUTS_ERR	0
ERR	Time ID is not initialized	Time correllations were not initialized. Computation not performed	XG_CFI_GEN_OSF_CREATE_TIME_INIT_ERR	1
ERR	Memory allocation error	Memory allocation error for the orbit change data structure Computation not performed	XG_CFI_GEN_OSF_CREATE_ALLOC_ERR	2
ERR	Wrong drift mode	Wrong drift mode flag value for charaterization of non-sun.synchronous orbits Computation not performed	XG_CFI_GEN_OSF_CREATE_DRIFT_MODE_ERR	3
ERR	Error calculating MLST drift	Error calculating MLST drift from inclination Computation not performed	XG_CFI_GEN_OSF_CREATE_DRIFT_CALC_ERR	4
ERR	Error calculating UTC of ANX	Error calculating the UTC time of the orbit ascending node Computation not performed	XG_CFI_GEN_OSF_CREATE_UTC_CALC_ERR	5

Table 9: Error messages of xg_gen_osf_create function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Error calculating TAI of ANX	Error calculating the TAI time of the orbit ascending node Computation not performed	XG_CFI_GEN_OSF_CREATE_TAI_CALC_ERR	6
ERR	Error calculating UT1 of ANX	Error calculating the UT1 time of the orbit ascending node Computation not performed	XG_CFI_GEN_OSF_CREATE_UT1_CALC_ERR	7
ERR	Error writing file to disk	Error writing the data structure to a file on disk Computation not performed	XG_CFI_GEN_OSF_CREATE_WRITE_ERR	8

7.2.6 Runtime performances

The following runtime performance has been measured.

Table 10: Runtime performances of xg_gen_osf_create function

Ultra Sparc [ms]
TBD

7.2.7 Executable Program

The **gen_osf_create** executable program can be called from a Unix shell as:

gen_osf_create **-sat** satellite_name
-orbit abs_orbit_number
-cyc cycle_number
-pha phase_number
-repcyc repeat_cycle (days)
-cyclen cycle_length (orbits)
-anx anx_long (deg)
{**-mlstdr** mlst_drift| **-inc** inclination}
-mlst mlst (decimal hours)
-date anx_date
[**-dir** dir_name] (current directory by default)
[**-osf** name of the orbit scenario file] (default: name generated automatically)
[**-flcl** file_class] (empty string by default)
[**-vers** version] (version=0 by default)
[**-fhsys** fh_system] (empty string by default)
[**-v**]
[**-xl_v**]
[**-xo_v**]
[**-xp_v**]
[**-xg_v**]
[**-help**]
[**-show**]
{(**-tai** TAI_time **-gps** GPS_time **-utc** UTC_time **-ut1** UT1_time) |
(**-tmod** time_model **-tfile** time_reference_data file **-trid** time_reference
{(**-tm0** time 0 **-tm1** time 1) | (**-orb0** orbit 0 **-orb1** orbit 1) })}

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.
- [**-xl_v**] option for EXPLORER_LIB Verbose mode.
- [**-xo_v**] option for EXPLORER_ORBIT Verbose mode.
- [**-xp_v**] option for EXPLORER_POINTING Verbose mode.
- [**-xg_v**] option for EXPLORER_GEN_FILES Verbose mode.
- [**-v**] option for Verbose mode for all libraries (default is Silent)for all libraries (default is Silent).
- [**-show**] displays the inputs of the function and the results.

- Possible values for *satellite_name*: ERS1, ERS2, ENVISAT, METOP1, METOP2, METOP3, CRYOSAT, ADM, GOCE, SMOS.
- 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_reference*: UNDEF, TAI, UTC, UT1, GPS.
- The last three lines of parameters are used to initialize the time references. In order to do this, only one set of parameters should be introduced:
  - TAI, GPS, UTC and UT1 input times
  - A file with time reference data, the time mode, the time reference name and the time range

Example:

```
gen_osf_create -sat CRYOSAT -orbit 1 -cyc 1 -pha 1 -repcyc 2
               -cyclen 29 -inc 92 -mlst 21 -date 790 -anx 130
               -dir ./gen_osf -osf mpl_orb_sc_at_302
               -tai -1100.1 -utc -1100.099595
               -ut1 -1100.0995914352 -gps -1100.0997801
```

7.3 xg_gen_osf_append_orbit_change

7.3.1 Overview

The **xg_gen_osf_append_orbit_change** CFI function appends an orbit change to an existing reference Orbit Scenario File (OSF). The user must provide in the calling interface the name of the existing OSF, the parameters describing the new orbit change and the output file name where the old OSF with the appended orbit change will be written. No output file is generated if the resulting orbit is discontinuous in terms of ascending node longitude, mean local solar time.

7.3.2 Calling interface

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

```
#include <explorer_gen_files.h>
{
      long sat_id;
      xl_time_id time_id;
      long abs_orbit_number,  repeat_cycle, cycle_length,
           drift_mode, phase_increment, version_number;
      double anx_long, inclination, mlst_drift, mlst;
      char input_filename[XG_MAX_LENGTH],
           output_dir[XG_MAX_LENGTH], output_filename[XG_MAX_LENGTH];
      char *file_class, *fh_system;
      long status, ierr[XG_ERR_VECTOR_MAX_LENGTH];
      status = xg_gen_osf_append_orbit_change (&sat_id, &time_id,
                                 &input_filename, &abs_orbit_number,
                                 &repeat_cycle, &cycle_length,
                                 &anx_long, &drift_mode,
                                 &inclination, &mlst_drift,
                                 &mlst, &phase_increment,
                                 output_dir, output_filename,
                                 file_class, &version_number,
                                 fh_system,
                                 ierr);

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

      status = xg_gen_osf_append_orbit_change_run (&run_id,
                                 &input_filename, &abs_orbit_number,
                                 &repeat_cycle, &cycle_length,
                                 &anx_long, &drift_mode,
```

```
                                        &inclination, &mlst_drift,
                                        &mlst, &phase_increment,
                                        output_dir, output_filename,
                                        file_class, &version_number,
                                        fh_system,
                                        ierr);
}
```

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

```
#include <explorer_gen_files.inc>

      INTEGER*4 SAT_ID, ABS_ORBIT_NUMBER, REPEAT_CYCLE, CYCLE_LENGTH,
      INTEGER*4 DRIFT_MODE, PHASE_INCREMENT, VERSION_NUMBER
      REAL*8 ANX_LONG, INCLINATION, MLST_DRIFT, MLST
      CHARACTER*XG_MAX_LENGTH INPUT_FILENAME
      CHARACTER*XG_MAX_LENGTH OUTPUT_DIR, OUTPUT_FILENAME
      CHAR*1 FILE_CLASS, FH_SYSTEM
      INTEGER*4 STATUS, IERR(XG_ERR_VECTOR_MAX_LENGTH)

      STATUS = XG_GEN_OSF_APPEND_ORBIT_CHANGE ( SAT_ID,
     &                                  INPUT_FILENAME, ABS_ORBIT_NUMBER,
     &                                  REPEAT_CYCLE, CYCLE_LENGTH,
     &                                  ANX_LONG, DRIFT_MODE,
     &                                  INCLINATION, MLST_DRIFT,
     &                                  MLST, PHASE_INCREMENT,
     &                                  OUTPUT_DIR, OUTPUT_FILENAME,
     &                                  FILE_CLASS, VERSION_NUMBER,
     &                                  FH_SYSTEM,
     &                                  IERR)
```

7.3.3 Input parameters

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

Table 11: Input parameters of xg_gen_osf_append_orbit_change 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.	-	-
input_filename	char*	-	Input OSF to which the orbit change is appended		
abs_orbit_number	long*	-	Absolute orbit number of the new orbit change	-	> abs orbit number in input OSF last orbit change
repeat_cycle	long*	-	Repeat cycle of the new reference orbit	days	>= 1
cycle_length	long*	-	Cycle length of the new reference orbit	orbits	>= 14
anx_long	double*	-	Requested orbit ascending node crossing longitude	deg	[-180, 180]
drift_mode	long*	-	Flag to select between drift in mean local solar time and inclination as input characterization of the reference orbit	-	[0,1]
inclination	double*	-	If *drift_mode* = *XO_NOSUNSYNC_INCLINATION* Inclination of the reference orbit	deg	[0,180]
mlst_drift	double*	-	If *drift_mode* = *XO_NOSUNSYNC_DRIFT* Drift in mean local solar time of the reference orbit: · MLST[N+1]=MLST[N]+MLST-drift	seconds/day	TBD
mlst	double*	-	Mean local solar time at ascending node	decimal hours	[0,24)
phase_increment	long*	-	If 1 then phase [N+1] = phase [N] + 1 If 0 then phase [N+1] = phase [N]	-	[0, 1]
output_dir	char*	-	Directory where the resulting OSF is written (if empty (i.e. ““), the current directory is used)	-	-

Table 11: Input parameters of xg_gen_osf_append_orbit_change function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
output_filename	char*	-	Output OSF name if empty (i.e. ""), the software will generate the filename according to file name specification pre-sented in [FORMATS]	-	-
file_class	char*	-	File class for output Orbit file	-	-
version_number	long*	-	Version number of output Orbit file	-	>= 0
fh_system	char*	-	System field of the output Orbit file fixed header	-	-

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

- Satellite ID: sat_id. See [GEN_SUM].
- Drift mode: mlst_drift. See [ORBIT_SUM].
- Phase increment: See current document table 2.

This CFI can append orbit changes for both sun-synchronous orbits and quasi-sun-synchronous orbits.

Use `drift_mode=XO_NOSUNSYNC_DRIFT` and `mlst_drift = 0.0` for a sun-synchronous orbit.

Use any other combination for the general case of quasi-sun-synchronous orbit.

7.3.4 Output parameters

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

Table 12: Output parameters of xg_gen_osf_append_orbit_change function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
ierr[XG_ERR_VECTOR_MAX_LENGTH]	long	all	Status vector	-	-

7.3.5 Warnings and errors

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

Table 13: Error messages of xg_gen_osf_append_orbit_change function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Wrong input values	Wrong value of one or more of the following input parameters: abs_orbit_number, repeat_cycle, cycle_length, mlst, phase_increment Computation not performed	XG_CFI_GEN_OSF_APPEND_INPUTS_ERR	0
ERR	Time ID is not initialized	Time correllations were not initialized. Computation not performed	XG_CFI_GEN_OSF_APPEND_TIME_INIT_ERR	1
ERR	Cannot read input OSF	Computation not performed	XG_CFI_GEN_OSF_APPEND_READ_IN_OSF_ERR	2
ERR	ANX long jump larger than allowed	Requested ANX long leads to an orbit discontinuity	XG_CFI_GEN_OSF_APPEND_ANX_LONG_ERR	3
ERR	MLST jump larger than allowed	Requested MLST leads to an orbit discontinuity	XG_CFI_GEN_OSF_APPEND_MLST_ERR	4
ERR	Wrong drift mode	Wrong drift mode flag value for charaterization of non-sun.synchronous orbits Computation not performed	XG_CFI_GEN_OSF_APPEND_DRIFT_MODE_ERR	5
ERR	Error calculating MLST drift	Error calculating MLST drift from inclination Computation not performed	XG_CFI_GEN_OSF_APPEND_DRIFT_CALC_ERR	6
ERR	Error calculating UTC of ANX	Error calculating the UTC time of the orbit ascending node Computation not performed	XG_CFI_GEN_OSF_APPEND_UTC_CALC_ERR	7

Table 13: Error messages of xg_gen_osf_append_orbit_change function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Error calculating TAI of ANX	Error calculating the TAI time of the orbit ascending node Computation not performed	XG_CFI_GEN_OSF_APPEND_TAI_CALC_ERR	8
ERR	Error calculating UT1 of ANX	Error calculating the UT1 time of the orbit ascending node Computation not performed	XG_CFI_GEN_OSF_APPEND_UT1_CALC_ERR	9
ERR	Error writing file to disk	Error writing the data structure to a file on disk Computation not performed	XG_CFI_GEN_OSF_APPEND_WRITE_ERR	10

7.3.6 Runtime performances

The following runtime performance has been measured.

Table 14: Runtime performances of xg_gen_osf_append_orbit_change function

Ultra Sparc [ms]
TBD

7.3.7 Executable Program

The **gen_osf_append_orbit_change** executable program can be called from a Unix shell as:

gen_osf_append_orbit_change **-sat** satellite_name
-inosf input_filename
-orbit abs_orbit_number
-repcyc repeat_cycle(days)
-cyclen cycle_length(orbits)
-anx anx_long(deg)
{ **-mlstdr** mlst_drift | **-inc** inclination }
-mlst mlst
[**-phinc**]
[**-dir** output_dir] (current directory by default)
[**-osf** output_filename] (default: name generated automatically)
[**-flcl** file_class] (empty string by default)
[**-vers** version] (version=0 by default)
[**-fhsys** fh_system] (empty string by default)
[**-v**]
[**-xl_v**]
[**-xo_v**]
[**-xp_v**]
[**-xg_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.
- [**-phinc**] option for phase_increment. Default value for phase_increment is XG_NO_PHASE_INCREMENT. When the option is written, phase_increment is XG_PHASE_INCREMENT.
- [**-xl_v**] option for EXPLORER_LIB Verbose mode.
- [**-xo_v**] option for EXPLORER_ORBIT Verbose mode.
- [**-xp_v**] option for EXPLORER_POINTING Verbose mode.

- [**-xg_v**] option for EXPLORER_GEN_FILES 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 *satellite_name*: ERS1, ERS2, ENVISAT, METOP1, METOP2, METOP3, CRYOSAT, ADM, GOCE, SMOS.
- 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_reference*: UNDEF, TAI, UTC, UT1, GPS.
- The last three lines of parameters are used to initialize the time references. In order to do this, 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)

Example:

```
gen_osf_append_orbit_change -sat CRYOSAT
     -inosf CS_TEST_MPL_ORBREF_20020301T122001_99999999T999999_0001.EEF
     -orbit 30 -repcyc 366 -cyclen 5344 -anx 129.9986 -mlst 20.90083
     -inc 92 -dir ./gen_osf -osf mpl_orb_sc_at_303
     -tai -1100.1 -utc -1100.099595
     -ut1 -1100.0995914352 -gps -1100.0997801
```

7.4 xg_gen_osf_change_repeat_cycle

7.4.1 Overview

Given a reference orbit from an existing OSF and a new target orbit (repeat cycle, cycle length, ascending node longitude and inclination or mean local solar time drift), the **xg_gen_osf_change_repeat_cycle** CFI function finds an optimum orbit change such that the target orbit can be reached from the found orbit change. This function will write a new OSF with the found orbit change appended to the content of the old OSF.

7.4.2 Calling interface

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

```
#include <explorer_gen_files.h>
{
      long sat_id;
      xl_time_id time_id;
      long abs_orbit_number, search_direction, repeat_cycle,
           cycle_length, drift_mode, phase_increment, version_number;
      double anx_long, inclination, mlst_drift;
      char input_filename[XG_MAX_LENGTH],
           output_dir[XG_MAX_LENGTH], output_filename[XG_MAX_LENGTH];
      char *file_class, *fh_system;
      long status, ierr[XG_ERR_VECTOR_MAX_LENGTH];
      status = xg_gen_osf_change_repeat_cycle (&sat_id, &time_id,
                                  &input_filename, &abs_orbit_number,
                                  &search_direction,
                                  &repeat_cycle, &cycle_length,
                                  &anx_long, &drift_mode,
                                  &inclination, &mlst_drift,
                                  &phase_increment,
                                  output_dir, output_filename,
                                  file_class, &version_number,
                                  fh_system,
                                  ierr);

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

      status = xg_gen_osf_change_repeat_cycle_run (&run_id,
                                  &input_filename, &abs_orbit_number,
                                  &search_direction,
```

```
                                        &repeat_cycle, &cycle_length,
                                        &anx_long, &drift_mode,
                                        &inclination, &mlst_drift,
                                        &phase_increment,
                                        output_dir, output_filename,
                                        file_class, &version_number,
                                        fh_system,
                                        ierr);
}
```

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

```
#include <explorer_gen_files.inc>

      INTEGER*4 SAT_ID, ABS_ORBIT_NUMBER, SEARCH_DIRECTION
      INTEGER*4 CYCLE_LENGTH, CYCLE_LENGTH, DRIFT_MODE, PHASE_INCREMENT
      REAL*8 ANX_LONG, INCLINATION, MLST_DRIFT
      CHARACTER*XG_MAX_LENGTH INPUT_FILENAME
      CHARACTER*XG_MAX_LENGTH OUTPUT_DIR, OUTPUT_FILENAME
      CHAR*1 FILE_CLASS, FH_SYSTEM
      INTEGER*4 STATUS, IERR(XG_ERR_VECTOR_MAX_LENGTH), VERSION_NUMBER

      STATUS = XG_GEN_OSF_CHANGE_REPEAT_CYCLE ( SAT_ID,
     &                                INPUT_FILENAME, ABS_ORBIT_NUMBER,
     &                                SEARCH_DIRECTION,
     &                                REPEAT_CYCLE, CYCLE_LENGTH,
     &                                ANX_LONG, DRIFT_MODE,
     &                                INCLINATION, MLST_DRIFT,
     &                                PHASE_INCREMENT,
     &                                OUTPUT_DIR, OUTPUT_FILENAME,
     &                                FILE_CLASS, VERSION_NUMBER,
     &                                FH_SYSTEM,
     &                                IERR)
```

7.4.3 Input parameters

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

Table 15: Input parameters of xg_gen_osf_change_repeat_cycle 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.	-	-
input_filename	char*	-	Input OSF to which the orbit change is appended		
abs_orbit_number	long*	-	Absolute orbit number from which the optimum transition search starts	-	> abs orbit number in input OSF last orbit change
search_direction	long*	-	Search for optimum transition after or before abs_orbit_number		{-1, 1}
repeat_cycle	long*	-	Repeat cycle of the new reference orbit	days	>= 1
cycle_length	long*	-	Cycle length of the new reference orbit	orbits	>= 14
anx_long	double*	-	Target orbit ascending node crossing longitude	deg	[-180, 180]
drift_mode	long*	-	Flag to select between drift in mean local solar time and inclination as input characterization of the reference orbit	-	[0,1]
inclination	double*	-	If *drift_mode* = *XO_NOSUNSYNC_INCLINATION* Inclination of the reference orbit	deg	[0,180]
mlst_drift	double*	-	If *drift_mode* = *XO_NOSUNSYNC_DRIFT* Drift in mean local solar time of the reference orbit: · MLST[N+1]=MLST[N]+MLST-drift	seconds/day	TBD
phase_increment	long*	-	If 1 then phase [N+1] = phase [N] + 1 If 0 then phase [N+1] = phase [N]	-	[0, 1]
output_dir	char*	-	Directory where the resulting OSF is written (if NULL, the current directory is used)	-	-

Table 15: Input parameters of xg_gen_osf_change_repeat_cycle function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
output_filename	char*	-	Output OSF name if empty (i.e. ""), the software will generate the filename according to file name specification pre-sented in [FORMATS]	-	-
file_class	char*	-	File class for output Orbit file	-	-
version_number	long*	-	Version number of output Orbit file	-	>= 0
fh_system	char*	-	System field of the output Orbit file fixed header	-	-

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

- Satellite ID: sat_id. See [GEN_SUM].
- Search direction: See current document table 2.
- Drift mode: mlst_drift. See [ORBIT_SUM].
- Phase increment: See current document table 2.

This CFI can append orbit changes for both sun-synchronous orbits and quasi-sun-synchronous orbits.

Use `drift_mode=XO_NOSUNSYNC_DRIFT` and `mlst_drift = 0.0` for a sun-synchronous orbit.

Use any other combination for the general case of quasi-sun-synchronous orbit.

7.4.4 Output parameters

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

Table 16: Output parameters of xg_gen_osf_change_repeat_cycle function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
ierr[XG_ERR_VECTOR_MAX_LENGTH]	long	all	Status vector	-	-

7.4.5 Warnings and errors

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

Table 17: Error messages of xg_gen_osf_change_repeat_cycle function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Wrong input values	Wrong value of one or more of the following input parameters: abs_orbit_number, search_direction, repeat_cycle, cycle_length, phase_increment Computation not performed	XG_CFI_GEN_OSF_CHANGE_INPUTS_ERR	0
ERR	Time ID is not initialized	Computation not performed	XG_CFI_GEN_OSF_CHANGE_TIME_INIT_ERR	1
ERR	Cannot read input OSF	Computation not performed	XG_CFI_GEN_OSF_CHANGE_READ_IN_OSF_ERR	2
ERR	Wrong drift mode	Wrong drift mode flag value for charaterization of non-sun.synchronous orbits Computation not performed	XG_CFI_GEN_OSF_CHANGE_DRIFT_MODE_ERR	3
ERR	Error calculating MLST drift	Error calculating MLST drift from inclination Computation not performed	XG_CFI_GEN_OSF_CHANGE_DRIFT_CALC_ERR	4
ERR	No transition found	No optimum transition found keeping orbit continuity Computation not performed	XG_CFI_GEN_OSF_CHANGE_NO_TRANSITION_ERR	5
ERR	Error calculating UTC of ANX	Error calculating the UTC time of the orbit ascending node Computation not performed	XG_CFI_GEN_OSF_CHANGE_UTC_CALC_ERR	6

Table 17: Error messages of xg_gen_osf_change_repeat_cycle function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Error calculating TAI of ANX	Error calculating the TAI time of the orbit ascending node Computation not performed	XG_CFI_GEN_OSF_CHANGE_TAI_CALC_ERR	7
ERR	Error calculating UT1 of ANX	Error calculating the UT1 time of the orbit ascending node Computation not performed	XG_CFI_GEN_OSF_CHANGE_UT1_CALC_ERR	8
ERR	Error writing file to disk	Error writing the data structure to a file on disk Computation not performed	XG_CFI_GEN_OSF_CHANGE_WRITE_ERR	9

7.4.6 Runtime performances

The following runtime performance has been measured.

Table 18: Runtime performances of xg_gen_osf_change_repeat_cycle function

Ultra Sparc [ms]
TBD

7.4.7 Executable Program

The **gen_osf_change_repeat_cycle** executable program can be called from a Unix shell as:

gen_osf_change_repeat_cycle **-sat** satellite_name
-inosf input_filename
-orbit abs_orbit_number
[**-back**]
-repcyc repeat_cycle(days)
-cyclen cycle_length(orbits)
-anx anx_long(deg)
{ **-mlstdr** mlst_drift | **-inc** inclination }
[**-phinc**]
[**-dir** output_dir] (current directory by default)
[**-osf** output_filename] (default: name generated automatically)
[**-flcl** file_class] (empty string by default)
[**-vers** version] (version=0 by default)
[**-fhsys** fh_system] (empty string by default)
[**-v**]
[**-xl_v**]
[**-xo_v**]
[**-xp_v**]
[**-xg_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.
- [**-back**] option for search_direction. Default value is XG_SEARCH_FORWARD. When the option is written, search_direction value is XG_SEARCH_BACKWARD.
- [**-phinc**] option for phase_increment. Default value is XG_NO_PHASE_INCREMENT. When the option is written, phase_increment value is XG_PHASE_INCREMENT.
- [**-xl_v**] option for EXPLORER_LIB Verbose mode.
- [**-xo_v**] option for EXPLORER_ORBIT Verbose mode.
- [**-xp_v**] option for EXPLORER_POINTING Verbose mode.
- [**-xg_v**] option for EXPLORER_GEN_FILES 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 *satellite_name*: ERS1, ERS2, ENVISAT, METOP1, METOP2, METOP3, CRYOSAT, ADM, GOCE, SMOS.
- 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_reference*: UNDEF, TAI, UTC, UT1, GPS.
- The last three lines of parameters are used to initialize the time references. In order to do this, 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)

Example:

```
gen_osf_change_repeat_cycle -sat CRYOSAT
     -inosf CS_TEST_MPL_ORBREF_20020301T122001_99999999T999999_0001.EEF
     -orbit 400 -repcyc 369 -cyclen 5344 -anx 286.524398  -inc 92
     -dir ./gen_osf -osf mpl_orb_sc_at_304
     -tai -1100.1 -utc -1100.099595
     -ut1 -1100.0995914352 -gps -1100.0997801
```

7.5 xg_gen_osf_add_drift_cycle

7.5.1 Overview

Given a reference orbit from an existing OSF, a new requested orbit with a particular ascending node longitude and an orbit for the manoeuvre, the **xg_gen_osf_add_drift_cycle** CFI function fits a repeat cycle/ cycle length between the manoeuvre orbit (drift start) and the requested orbit (drift stop) such that the longitude of the ascending node at the drift stop orbit be the one requested.

The drift orbit is constrained by a maximum altitude difference with respect to the reference orbit.

Furthermore, if the reference orbit is sun-synchronous, the drift orbit shall also be sun-synchronous; but if the reference orbit is not sun-synchronous, the drift orbit shall keep the inclination constant.

This CFI appends two orbit changes to the existing OSF:

- The first one for the drift manoeuvre
- The second one for restoring the old reference orbit characteristics at the requested ascending node longitude

7.5.2 .Calling interface

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

```
#include <explorer_gen_files.h>
{
      long sat_id;
      xl_time_id time_id;
      long drift_start_orbit, drift_stop_orbit,
           phase_inc_start, phase_inc_stop, version_number;
      double drift_stop_anx_long, max_altitude_change;
      char input_filename[XG_MAX_LENGTH],
           output_dir[XG_MAX_LENGTH], output_filename[XG_MAX_LENGTH];
      char *file_class, *fh_system;
      long status, ierr[XG_ERR_VECTOR_MAX_LENGTH];
      status = xg_gen_osf_add_drift_cycle (&sat_id, &time_id,
                                  &input_filename,
                                  &drift_start_orbit,
                                  &drift_stop_orbit,
                                  &drift_stop_anx_long,
                                  &max_altitude_change,
                                  &phase_inc_start, &phase_inc_stop,
                                  output_dir, output_filename,
                                  file_class, &version_number,
                                  fh_system,
                                  ierr);
```

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

        status = xg_gen_osf_add_drift_cycle_run (&run_id,
                                        &input_filename,
                                        &drift_start_orbit,
                                        &drift_stop_orbit,
                                        &drift_stop_anx_long,
                                        &max_altitude_change,
                                        &phase_inc_start, &phase_inc_stop,
                                        output_dir, output_filename,
                                        file_class, &version_number,
                                        fh_system,
                                        ierr);
}
```

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

```
#include <explorer_gen_files.inc>

        INTEGER*4 SAT_ID, DRIFT_START_ORBIT, DRIFT_STOP_ORBIT,
        INTEGER*4 PHASE_INC_START, PHASE_INC_STOP, VERSION_NUMBER
        REAL*8 DRIFT_STOP_ANX_LONG, MAX_ALTITUDE_CHANGE
        CHARACTER*XG_MAX_LENGTH INPUT_FILENAME
        CHARACTER*XG_MAX_LENGTH OUTPUT_DIR, OUTPUT_FILENAME
        CHAR*1 FILE_CLASS, FH_SYSTEM
        INTEGER*4 STATUS, IERR(XG_ERR_VECTOR_MAX_LENGTH)

        STATUS = XG_GEN_OSF_ADD_DRIFT_CYCLE ( SAT_ID,
      &                                 INPUT_FILENAME,
      &                                 DRIFT_START_ORBIT,
      &                                 DRIFT_STOP_ORBIT,
      &                                 DRIFT_STOP_ANX_LONG,
      &                                 MAX_ALTITUDE_CHANGE,
      &                                 PHASE_INC_START, PHASE_INC_STOP,
      &                                 OUTPUT_DIR, OUTPUT_FILENAME,
      &                                 FILE_CLASS, VERSION_NUMBER,
      &                                 FH_SYSTEM,
      &                                 IERR)
```

7.5.3 Input parameters

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

Table 19: Input parameters of xg_gen_osf_add_drift_cycle 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.	-	-
input_filename	char*	-	Input OSF to which the orbit changes are appended		
drift_start_orbit	long*	-	Absolute orbit number at the drift start	-	> abs orbit number in input OSF last orbit change
drift_stop_orbit	long*	-	Absolute orbit number at the drift stop	-	> drift_start_orbit
drift_stop_anx_long	double*	-	Drift stop orbit ascending node crossing longitude	deg	[-180, 180]
max_altitude_change	double*	-	Maximum variation in altitude between the reference orbit and the drift orbit	m	

phase_inc_start	long*	-	Phase increment at drift start
			If 1 then
			phase [N+1] = phase [N] + 1
			If 0 then
			phase [N+1] = phase [N]
				-	[0, 1]

phase_inc_stop	long*	-	Phase increment at drift stop
			If 1 then
			phase [N+1] = phase [N] + 1
			If 0 then
			phase [N+1] = phase [N]
				-	[0, 1]

output_dir	char*	-	Directory where the resulting OSF is written (if empty (i.e. ""), the current directory is used)	-	-
output_filename	char*	-	Output OSF name if empty (i.e. ""), the software will generate the filename according to file name specification pre-sented in [FORMATS]	-	-
file_class	char*	-	File class for output Orbit file	-	-
version_number	long*	-	Version number of output Orbit file	-	>= 0
fh_system	char*	-	System field of the output Orbit file fixed header	-	-

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

- Satellite ID: sat_id. See [GEN_SUM].
- Search direction: See current document table 2.
- Drift mode: mlst_drift. See [ORBIT_SUM].
- Phase increment: See current document table 2.

7.5.4 Output parameters

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

Table 20: Output parameters of xg_gen_osf_add_drift_cycle function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
ierr[XG_ERR_VECTOR_MAX_LENGTH]	long	all	Status vector	-	-

7.5.5 Warnings and errors

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

Table 21: Error messages of xg_gen_osf_add_drift_cycle function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Wrong input values	Wrong value of one or more of the following input parameters: drift_start_orbit, drift_stop_orbit, phase_inc_start, phase_inc_stop, Computation not performed	XG_CFI_GEN_OSF_DRIFT_INPUTS_ERR	0
ERR	Time ID is not initialized	Computation not performed	XG_CFI_GEN_OSF_DRIFT_TIME_INIT_ERR	1
ERR	Cannot read input OSF	Computation not performed	XG_CFI_GEN_OSF_DRIFT_READ_IN_OSF_ERR	2
ERR	No drift orbit necessary	Computation not performed	XG_CFI_GEN_OSF_DRIFT_NO_ADD_ERR	3

Table 21: Error messages of xg_gen_osf_add_drift_cycle function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Error calculating inclination	Error calculating inclination for a no sun-synchronous orbit in order to keep inclination constant during the drift phase Computation not performed	XG_CFI_GEN_OSF_DRIFT_INCL_CALC_ERR	4
ERR	No drift orbit found	No drift orbit has been found that matches the drift start and stop ANX longitude Computation not performed	XG_CFI_GEN_OSF_DRIFT_NOT_FOUND_ERR	5
ERR	Error calculating UTC of ANX	Error calculating the UTC time of the orbit ascending node Computation not performed	XG_CFI_GEN_OSF_DRIFT_UTC_CALC_ERR	6
ERR	Error calculating TAI of ANX	Error calculating the TAI time of the orbit ascending node Computation not performed	XG_CFI_GEN_OSF_DRIFT_TAI_CALC_ERR	7
ERR	Error calculating UT1 of ANX	Error calculating the UT1 time of the orbit ascending node Computation not performed	XG_CFI_GEN_OSF_DRIFT_UT1_CALC_ERR	8
ERR	Error writing file to disk	Error writing the data structure to a file on disk Computation not performed	XG_CFI_GEN_OSF_DRIFT_WRITE_ERR	9

7.5.6 Runtime performances

The following runtime performance has been measured.

Table 22: Runtime performances of xg_gen_osf_add_drift_cycle function

Ultra Sparc [ms]
TBD

7.5.7 Executable Program

The **gen_osf_add_drift_cycle** executable program can be called from a Unix shell as:

gen_osf_add_drift_cycle **-sat** satellite_name
-inosf input_filename
-drorb0 drift_start_orbit
-drorb1 drift_stop_orbit
-anx drift_stop_anx_long (deg)
-alt max_altitude_change (m)
[**-phinc0**]
[**-phinc1**]
[**-dir** output_dir] (current directory by default)
[**-osf** output_filename] (default: name generated automatically)
[**-flcl** file_class] (empty string by default)
[**-vers** version] (version=0 by default)
[**-fhsys** fh_system] (empty string by default)
[**-v**]
[**-xl_v**]
[**-xo_v**]
[**-xp_v**]
[**-xg_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.
- [**-phinc0**] option for phase_inc_start. Default value is XG_NO_PHASE_INCREMENT. When the option is written, phase_inc_start value is XG_PHASE_INCREMENT.
- [**-phinc1**] option for phase_inc_stop. Default value is XG_NO_PHASE_INCREMENT. When the option is written, phase_inc_stop value is XG_PHASE_INCREMENT.
- [**-xl_v**] option for EXPLORER_LIB Verbose mode.
- [**-xo_v**] option for EXPLORER_ORBIT Verbose mode.
- [**-xp_v**] option for EXPLORER_POINTING Verbose mode.
- [**-xg_v**] option for EXPLORER_GEN_FILES 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 *satellite_name*: ERS1, ERS2, ENVISAT, METOP1, METOP2, METOP3, CRYOSAT, ADM, GOCE, SMOS.
- 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_reference*: UNDEF, TAI, UTC, UT1, GPS.
- The last three lines of parameters are used to initialize the time references. In order to do this, 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)

Example:

```
gen_osf_add_drift_cycle -sat CRYOSAT
-inosf CS_TEST_MPL_ORBREF_20020301T122001_99999999T999999_0001.EEF
-drorb0 30 -drorb1 2702 -anx 310 -alt 15000 -dir ./gen_osf
-osf mpl_orb_sc_at_305
-tai -1100.1 -utc -1100.099595 -ut1 -1100.0995914352 -gps -1100.0997801
```

7.6 xg_gen_rof

7.6.1 Overview

The **xg_gen_rof** CFI function creates a Restituted Orbit File (ROF) using as input one of the following reference file types:

- Orbit Scenario File
- FOS Predicted Orbit File
- DORIS Navigator File
- FOS Restituted Orbit File
- DORIS Preliminary Orbit File
- DORIS Precise Orbit FileTime of the ascending crossing node (TAI, UTC and UT1)

The accepted output file types are:

- FOS Restituted Orbit File
- DORIS Preliminary Orbit File
- DORIS Precise Orbit FileTime

The time interval between consecutive OSVs can be selected by the user by means of a parameter in the calling interface. A flag for precise location of OSVs at “integer intervals” (e.g. every exact minute) is also available. If the reference file and the Restituted Orbit File contain OSVs at the same time, these OSVs will be identical.
Note: when using an OSF or Predicted Orbit file, the maximum time interval within the output Restituted orbit file is limited to 2 orbital periods before and after the middle point of the user requested time range.

7.6.2 Calling interface

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

```
#include <explorer_gen_files.h>
{
      long sat_id;
      xl_time_id time_id;
      long   time_init, time_ref, start_orbit, stop_orbit,
             ref_filetype, rof_filetype, osv_precise, version_number;
      double start_time, stop_time, osv_interval;
      char   reference_file[XG_MAX_LENGTH], output_dir[XG_MAX_LENGTH],
             rof_filename[XG_MAX_LENGTH];
      char  *file_class, *fh_system;
      long   status, ierr[XG_ERR_VECTOR_MAX_LENGTH];
      status = long xg_gen_rof(&sat_id, &time_id, &time_init,
                               &time_ref, &start_time,&stop_time,
                               &start_orbit, &stop_orbit,
                               &osv_interval, &osv_precise,
```

```
                                &ref_filetype, reference_file,
                                &rof_filetype, output_dir, rof_filename,
                                file_class, &version_number, fh_system,
                                /* output */
                                ierr);


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

        status = long xg_gen_rof_run(&run_id, &time_init, &time_ref,
                                &start_time,&stop_time,
                                &start_orbit, &stop_orbit,
                                &osv_interval, &osv_precise,
                                &ref_filetype, reference_file,
                                &rof_filetype, output_dir, rof_filename,
                                file_class, &version_number, fh_system,
                                /* output */
                                ierr);
}
```

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

```
#include <explorer_gen_files.inc>

        INTEGER*4 SAT_ID, TIME_INIT, TIME_REF, START_ORBIT, STOP_ORBIT
        INTEGER*4 OSV_PRECISE, REF_FILETYPE, ROF_FILETYPE, VERSION_NUMBER
        REAL*8 START_TIME, STOP_TIME, OSV_INTERVAL
        CHAR*1 OUTPUT_DIR(XG_MAX_LENGTH), ROF_FILENAME(XG_MAX_LENGTH)
        CHAR*1 REFERENCE_FILENAME(XG_MAX_LENGTH), FILE_CLASS, FH_SYSTEM
        INTEGER*4 STATUS, IERR(XG_ERR_VECTOR_MAX_LENGTH)

        STATUS = XG_GEN_ROF   ( SAT_ID, TIME_INIT,
      &                         TIME_REF, START_TIME,
      &                         STOP_TIME, START_ORBIT,
      &                         STOP_ORBIT, OSV_INTERVAL, OSV_PRECISE
      &                         REF_FILETYPE, REFERENCE_FILENAME,
      &                         ROF_FILETYPE,
      &                         OUTPUT_DIR, ROF_FILENAME,
      &                         FILE_CLASS, VERSION_NUMBER,FH_SYSTEM,
      &                         IERR)
```

7.6.3 Input parameters

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

Table 23: Input parameters of xg_gen_rof 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. NOTE:Time correlations are only required if the input reference orbit file can not initialise them.	-	-
time_init	long*	-	Flag for selecting the time range of the initialisation.	-	Select either: · XO_SEL_ORBIT · XO_SEL_TIME
time_ref	long*	-	Time reference ID (see note in the ref_filetype field)	-	Complete
start_time	double*	-	Processing time corresponding to the beginning of the required interval	Decimal days, MJD2000	[-18262.0,36524.0]
stop_time	double*	-	Processing time corresponding to the end of the required interval	Decimal days, MJD2000	[-18262.0,36524.0]
start_orbit	long*	-	Orbit number corresponding to the beginning of the required interval	orbits	>= 1
stop_orbit	long*	-	Orbit number corresponding to the end of the required interval	orbits	>= 1
osv_interval	double*	-	Interval between consecutive state vector. This parameter should be coherent with the osv_precise flag (see below). If osv_precise is set to: • XG_OSV_PRECISE_MINUTE: osv will be forced to be a multiple of 60 seconds. • XG_OSV_PRECISE_TEN_SECONDS: osv will be forced to be a multiple of 10 seconds.	secs	>=0
osv_precise	long*	-	Flag to indicate if state vectors should be placed at exact time locations	-	Complete

Table 23: Input parameters of xg_gen_rof function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
ref_filetype	long*	-	File type of the input reference file. (Note: When generating a ROF file from a DORIS NAVIGATOR file, the input times should be expresed in UTC)	-	Complete
reference_filename	char*	-	Reference File name	-	
rof_filetype	long*	-	File type of the output reference file	-	XG_REF_FILETYPE_ROF XG_REF_FILETYPE_DORIS_PREM XG_REF_FILETYPE_DORIS_PREC
output_dir	char*	-	Directory where the resulting ROF is written (if NULL, the current directory is used)	-	-
rof_filename	char*	-	Output ROF name if empty (i.e. ""), the software will generate the filename according to file name specification presented in [FORMATS]	-	-
file_class	char*	-	File class for output Restituted file	-	-
version_number	long*	-	Version number of output Restituted file	-	>= 0
fh_system	char*	-	System field of the output Restituted file fixed header	-	-

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 initialisation: time_init. See [ORBIT_SUM].
- Time reference: time_ref. See [ORBIT_SUM].
- OSV precise: osv_precise. See this SUM.
- File type: ref_filetype and rof_filetype. See this SUM.

7.6.4 Output parameters

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

Table 24: Output parameters of xg_gen_rof function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
ierr[XG_ERR_VECTOR_MAX_LENGTH]	long	all	Status vector	-	-

7.6.5 Warnings and errors

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

Table 25: Error messages of xg_gen_rof function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Wrong satellite flag	Computation not performed	XG_CFI_GEN_ROF_WRONG_SAT_ID_ERR	0
ERR	Wrong input flag	Computation not performed	XG_CFI_GEN_ROF_WRONG_FLAG_ERR	1
ERR	Time ID is not initialized	Computation not performed	XG_CFI_GEN_ROF_TIME_INIT_ERR	2
ERR	Could not initialise the time reference	Computation not performed	XG_CFI_GEN_ROF_TIME_INITIALIZATION_ERR	3
ERR	Cannot create in-memory XML tree	Computation not performed	XG_CFI_GEN_ROF_CREATE_TREE_ERR	4
ERR	Cannot create root element	Computation not performed	XG_CFI_GEN_ROF_CREATE_ROOT_ERR	5
ERR	Cannot add XML node to tree	Computation not performed	XG_CFI_GEN_ROF_CREATE_NODE_ERR	6
ERR	Cannot initialise orbit ID	Computation not performed	XG_CFI_GEN_ROF_ORBIT_INIT_FILE_ERR	7
ERR	Cannot initialise the propagator	Computation not performed	XG_CFI_GEN_ROF_PROPAG_INIT_ERR	8
ERR	Could not perform a time <-> orbit transformation	Computation not performed	XG_CFI_GEN_ROF_TIME_ORBIT_ERR	9

Table 25: Error messages of xg_gen_rof function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Cannot initialise interpolation	Computation not performed	XG_CFI_GEN_ROF_INTERPOL_INIT_ERR	10
ERR	Wrong interpol initialisation	Computation not performed	XG_CFI_GEN_ROF_INTERNAL1_ERR	11
ERR	Cannot calculate state vector	Computation not performed	XG_CFI_GEN_ROF_CALCULATING_STATE_VECTOR_ERR	12
ERR	Cannot convert time to processing format	Computation not performed	XG_CFI_GEN_ROF_TIME_ERR	13
ERR	Cannot convert time from processing to external	Computation not performed	XG_CFI_GEN_ROF_TIME_TO_EXTERNAL_ERR	14
ERR	Cannot write state vector in XML file	Computation not performed	XG_CFI_GEN_ROF_WRITE_SV_ERR	15
ERR	Error freeing memory	Computation not performed	XG_CFI_GEN_ROF_CLOSE_ERR	16
ERR	Cannot write ROF XML file	Computation not performed	XG_CFI_GEN_ROF_FILE_ERR	17
ERR	Error creating the fixed header	Computation not performed	XG_CFI_GEN_ROF_CREATE_HEADER_ERR	18
ERR	Error opening the control file	Computation not performed	XG_CFI_GEN_ROF_CONTROL_FILE_OPEN_ERR	19
ERR	Control file does not cover the interval required	Computation not performed	XG_CFI_GEN_ROF_INVALID_CONTROL_FILE_ERR	20
ERR	Error reading the control file	Computation not performed	XG_CFI_GEN_ROF_CONTROL_FILE_READ_ERR	21
ERR	Error opening auxiliary file in the current directory	Computation not performed	XG_CFI_GEN_ROF_AUX_FILE_OPEN_ERR	22
ERR	Cannot correct state vector	Computation not performed	XG_CFI_GEN_ROF_CORRECT_OSV_ERR	23
ERR	Error writing ascii data in DORIS navigator file	Computation not performed	XG_CFI_GEN_ROF_ASCII_DATA_ERR	24
ERR	Error getting the parameter for the XML Header File	Computation not performed	XG_CFI_GEN_ROF_COMPUTE_HEADER_ERR	25
ERR	Error creating the XML Header File	Computation not performed	XG_CFI_GEN_ROF_CREATE_HEADER_FILE_ERR	26
ERR	OSV interval is not compatible with OSV Precise flag. The OSV Interval will be set to %f seconds.	Computation performed with a different value for the osv_interval	XG_CFI_GEN_ROF_WRONG_INTERVAL_WARN	27

7.6.6 Runtime performances

The following runtime performance has been measured.

Table 26: Runtime performances of xg_gen_rof function

Ultra Sparc [ms]
TBD

7.6.7 Executable Program

The **gen_rof** executable program can be called from a Unix shell as:

gen_rof **-sat** satellite_name
-tref time_ref
{ **-tstart** start_time **-tstop** stop_time (decimal days) |
-tastart start_time **-tastop** stop_time (CCSDSA format) |
-ostart start_orbit **-ostop** stop_orbit (orbits) }
-osvint osv_interval
[**-osvpre**]
-reftyp ref_file_type
-ref reference_file
-roftyp rof_file_type
[**-dir** output_dir] (current directory by default)
[**-rof** output_filename] (default: name generated automatically)
[**-flcl** file_class] (empty string by default)
[**-vers** version] (version=0 by default)
[**-fhsys** fh_system] (empty string by default)
[**-v**]
[**-xl_v**]
[**-xo_v**]
[**-xp_v**]
[**-xg_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.
- [**-osvpre**] option for osv_precise.Default value is XG_OSV_PRECISE_NO. When the option is written, ovs_precise value is XG_OSV_PRECISE_MINUTE.
- [**-xl_v**] option for EXPLORER_LIB Verbose mode.
- [**-xo_v**] option for EXPLORER_ORBIT Verbose mode.
- [**-xp_v**] option for EXPLORER_POINTING Verbose mode.
- [**-xg_v**] option for EXPLORER_GEN_FILES 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 *satellite_name*: ERS1, ERS2, ENVISAT, METOP1, METOP2, METOP3, CRYOSAT, ADM, GOCE, SMOS.
- 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 *ref_file_type*: OSF, POF, DORISNAV, ROF, DORISPREM, DORISPREC.
- Possible values for *rof_file_type*: ROF, DORISPREM, DORISPREC.
- Possible values for *time_ref* and *time_reference*: UNDEF, TAI, UTC, UT1, GPS.
- Time references need to be initialized only when using OSF as the type of the input reference file. The inputs needed for this issue are provided in the last three lines of parameters. 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)

Example:

```
gen_rof   -sat CRYOSAT -tref TAI -ostart 1000 -ostop 1001
          -osvint 300 -reftyp OSF
          -ref CS_TEST_MPL_ORBREF_20020301T122001_99999999T999999_0001.EEF
          -roftyp ROF -dir ./gen_rof/ -rof orb_res_file_at_306
          -tmod FOS_PREDICTED -tfile ./data/test.fpo -trid TAI
          -tm0 0 -tm1 10000
```

7.7 xg_gen_rof_prototype

7.7.1 Overview

The **xg_gen_rof_prototype** CFI function creates a Restituted Orbit File (ROF) using the following input parameters:

- Date (processing time) and orbit
- Longitude of the ascending node,
- Satellite Repeat Cycle and Cycle Length
- Mean local solar time at ascending node
- Drift of mean local solar time or the inclination

The time interval between consecutive OSVs can be selected by the user by means of a parameter in the calling interface.

7.7.2 Calling interface

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

```
#include <explorer_gen_files.h>
{
      long sat_id;
      xl_time_id time_id;
      long propag_model, time_ref, time_init_mode;
      long orbit0, drift_mode, irep, icyc, start_orbit, stop_orbit;
      double time0, start_time, stop_orbit, osv_interval;
      double ascmlst_drift, inclination, rlong, ascmlst;
      char   output_dir[XG_MAX_LENGTH], rof_filename[XG_MAX_LENGTH];
      char  *file_class, *fh_system;
      long status, ierr[XG_ERR_VECTOR_MAX_LENGTH], version_number;
      status = xg_gen_rof_prototype (&sat_id, &time_id,
                                     &propag_model, &time_ref,
                                     &time0, &orbit0, &time_init_mode,
                                     &start_time, &start_orbit
                                     &stop_time, &stop_orbit,
                                     &drift_mode,
                                     &ascmlst_drift, &inclination,
                                     &irep, &icyc, &rlong, &ascmlst,
                                     &osv_interval
                                     output_dir,rof_filename,
                                     file_class, &version_number,
                                     fh_system,
                                    /* output */
                                    ierr);
```

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

      status = xg_gen_rof_prototype_run (&run_id,
                                         &propag_model, &time_ref,
                                         &time0, &orbit0, &time_init_mode,
                                         &start_time, &start_orbit
                                         &stop_time, &stop_orbit,
                                         &drift_mode,
                                         &ascmlst_drift, &inclination,
                                         &irep, &icyc, &rlong, &ascmlst,
                                         &osv_interval
                                         output_dir,rof_filename,
                                         file_class, &version_number,
                                         fh_system,
                                        /* output */
                                       ierr);
}
```

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

```
#include <explorer_gen_files.inc>

      INTEGER*4 SAT_ID, PROPAG_MODEL, TIME_REF, TIME_INIT_MODE
      INTEGER*4 DRIFT_MODE, IREP, ICYC, ORBIT0, START_ORBIT, STOP_ORBIT
      REAL*8 TIME0, START_TIME, STOP_TIME
      REAL*8 ASCMLST_DRIFT, INCLINATION, RLONG, ASCMLST, OSV_INTERVAL
      CHAR*1 OUTPUT_DIR(XG_MAX_LENGTH), ROF_FILENAME(XG_MAX_LENGTH)
      CHAR*1 FILE_CLASS, FH_SYSTEM
      INTEGER*4 STATUS,IERR(XO_NUM_ERR_PROPAG_INIT_DEF),VERSION_NUMBER

      STATUS = XG_GEN_ROF_PROTOTYPE
     &                               (SAT_ID, PROPAG_MODEL,
     &                                TIME_REF, TIME0, ORBIT0,
     &                                TIME_INIT_MODE, START_TIME,
     &                                START_ORBIT, STOP_TIME, STOP_ORBIT,
     &                                DRIFT_MODE, ASCMLST_DRIFT,
     &                                INCLINATION, IREP, ICYC, RLONG,
     &                                ASCMLST, OSV_INTERVAL
     &                                OUTPUT_DIR, ROF_FILENAME,
     &                                FILE_CLASS, VERSION_NUMBER,
     &                                FH_SYSTEM,
     &                                IERR)
```

7.7.3 Input parameters

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

Table 27: Input parameters of xg_gen_rof_prototype 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. NOTE:Time correlations are only required if the input reference orbit file can not initialise them.	-	-
propag_model	long*	-	Propagation model ID	-	Complete
time_ref	long*	-	Time reference ID	-	Complete
time0	double*	-	Reference time	Decimal days (Processing format)	[-18262.0,36524.0]
orbit0	long*	-	Absolute orbit number of the reference orbit	-	>= 0
time_init_mode	long*	-	Flag for selecting the time range of the initialisation.	-	Select either: · XO_SEL_ORBIT · XO_SEL_TIME
start_time	double*	-	Processing time corresponding to the beginning of the required interval	Decimal days, MJD2000	[-18262.0,36524.0]
start_orbit	long*	-	Orbit number corresponding to the beginning of the required interval	orbits	>= 1
stop_time	double*	-	Processing time corresponding to the end of the required interval	Decimal days, MJD2000	[-18262.0,36524.0]
stop_orbit	long*	-	Orbit number corresponding to the end of the required interval	orbits	>= 1
drift_mode	long*	-	Flag to select between drift in mean local solar time and inclination as input characterization of the reference orbit	-	Complete
ascmlst_drift	double*	-	If *drift_mode* = *XO_NOSUNSYNC_MLST* Drift in mean local solar time of the reference orbit	seconds/day	TBD
inclination	double*	-	If *drift_mode* = *XO_NOSUNSYNC_INCLINATION* Inclination of the reference orbit	deg	[0,180]

Table 27: Input parameters of xg_gen_rof_prototype function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
irep	long *	-	Repeat cycle of the reference orbit The actual repeat cycle is calculated as per definition included in [MCD].	days	> 0
icyc	long *	-	Cycle length of the reference orbit	orbits	> 0
rlong	double*	-	Geocentric longitude of the [Earth fixed] ascending node (Earth fixed CS)	deg	[0,360)
ascmlst	double*	-	Mean local solar time at ascending node	seconds	[0,86400)
osv_interval	double*	-	Interval between consecutive state vector	secs	>=0
output_dir	char*	-	Directory where the resulting ROF is written (if NULL, the current directory is used)	-	-
rof_filename	char*	-	Output ROF name if empty (i.e. ""), the software will generate the filename according to file name specification presented in [FORMATS]	-	-
file_class	char*	-	File class for output Restituted file	-	-
version_number	long*	-	Version number of output Restituted file	-	>= 0
fh_system	char*	-	System field of the output Restituted file fixed header	-	-

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 initialisation: time_init. See [ORBIT_SUM].
- Time reference: time_ref. See [ORBIT_SUM].
- Drift Mode: drift_mode. See [ORBIT_SUM].

7.7.4 Output parameters

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

Table 28: Output parameters of xg_gen_rof_prototype function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
ierr[XG_ERR_VECTOR_MAX_LENGTH]	long	all	Status vector	-	-

7.7.5 Warnings and errors

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

Table 29: Error messages of xg_gen_rof_prototype function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Wrong satellite flag	Computation not performed	XG_CFI_GEN_ROF_PROTOTYPE_WRONG_SAT_ID_ERR	0
ERR	Time ID is not initialized	Computation not performed	XG_CFI_GEN_ROF_PROTOTYPE_TIME_ID_ERR	1
ERR	Wrong input flag	Computation not performed	XG_CFI_GEN_ROF_PROTOTYPE_WRONG_FLAG_ERR	2
ERR	Cannot create in-memory XML tree	Computation not performed	XG_CFI_GEN_ROF_PROTOTYPE_CREATE_TREE_ERR	3
ERR	Cannot create root element	Computation not performed	XG_CFI_GEN_ROF_PROTOTYPE_CREATE_ROOT_ERR	4
ERR	Cannot add XML node to tree	Computation not performed	XG_CFI_GEN_ROF_PROTOTYPE_CREATE_NODE_ERR	5
ERR	Cannot initialise propagator	Computation not performed	XG_CFI_GEN_ROF_PROTOTYPE_PROPAG_INIT_DEF_ERR	6

Table 29: Error messages of xg_gen_rof_prototype function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Cannot calculate state vector	Computation not performed	XG_CFI_GEN_ROF_PROTOTYPE_CALCULATING_STATE_VECTOR_ERR	7
ERR	Cannot convert time in processing reference	Computation not performed	XG_CFI_GEN_ROF_PROTOTYPE_TIME_ERR	8
ERR	Cannot convert time from processing to external	Computation not performed	XG_CFI_GEN_ROF_PROTOTYPE_TIME_TO_EXTERNAL_ERR	9
ERR	Cannot write state vector in XML file	Computation not performed	XG_CFI_GEN_ROF_PROTOTYPE_WRITE_SV_ERR	10
ERR	Error freeing memory	Computation not performed	XG_CFI_GEN_ROF_PROTOTYPE_CLOSE_ERR	11
ERR	Cannot write ROF XML file	Computation not performed	XG_CFI_GEN_ROF_PROTOTYPE_FILE_ERR	12
ERR	Error creating the fixed header	Computation not performed	XG_CFI_GEN_ROF_PROTOTYPE_CREATE_HEADER_ERR	13

7.7.6 Runtime performances

The following runtime performance has been measured.

Table 30: Runtime performances of xg_gen_rof_prototype function

Ultra Sparc [ms]
TBD

7.8 xg_gen_pof

7.8.1 Overview

The **xg_gen_pof** CFI function creates a Predicted Orbit File (POF) with one state vector per orbit using as input one of the following reference file types:

- Orbit Scenario File
- FOS Predicted Orbit File
- DORIS Navigator File
- FOS Restituted Orbit File
- DORIS Preliminary Orbit File
- DORIS Precise Orbit FileTime of the ascending crossing node (TAI, UTC and UT1)

The location of the state vector within the orbit can be selected by the user by means of a parameter in the calling interface. If the reference file and the Predicted Orbit File contain OSVs at the same time, these OSVs will be identical.

7.8.2 Calling interface

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

```
#include <explorer_gen_files.h>
{
      long    sat_id;
      xl_time_id time_id;
      long    time_init, time_ref, start_orbit, stop_orbit,
              ref_filetype, pof_filetype, version_number;
      double start_time, stop_time, osv_location;
      char    reference_file[XG_MAX_LENGTH], output_dir[XG_MAX_LENGTH],
              pof_filename[XG_MAX_LENGTH];
      char   *file_class, *fh_system;
      long    status, ierr[XG_ERR_VECTOR_MAX_LENGTH];
      status = long xg_gen_pof(&sat_id, &time_id,
                               &time_init, &time_ref,
                               &start_time,&stop_time,
                               &start_orbit, &stop_orbit, &osv_location,
                               &ref_filetype, reference_file,
                              &pof_filetype, output_dir,pof_filename,
                               file_class, &version_number, fh_system,
                              /* output */
                             ierr);
```

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

      status = long xg_gen_pof_run(&run_id,
                                   &time_init, &time_ref,
                                   &start_time,&stop_time,
                                   &start_orbit, &stop_orbit, &osv_location,
                                   &ref_filetype, reference_file,
                                  &pof_filetype, output_dir,pof_filename,
                                   file_class, &version_number, fh_system,
                                  /* output */
                                  ierr);
}
```

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

```
#include <explorer_gen_files.inc>

      INTEGER*4 SAT_ID, TIME_INIT, TIME_REF, START_ORBIT, STOP_ORBIT
      INTEGER*4 REF_FILETYPE, POF_FILETYPE, VERSION_NUMBER
      REAL*8 START_TIME, STOP_TIME, OSV_LOCATION
      CHAR*1 OUTPUT_DIR(XG_MAX_LENGTH), POF_FILENAME(XG_MAX_LENGTH)
      CHAR*1 REFERENCE_FILENAME(XG_MAX_LENGTH)
      CHAR*1 FILE_CLASS, FH_SYSTEM
      INTEGER*4 STATUS, IERR(XG_ERR_VECTOR_MAX_LENGTH)

      STATUS = XG_GEN_POF   ( SAT_ID, TIME_INIT,
     &                         TIME_REF, START_TIME,
     &                         STOP_TIME, START_ORBIT,
     &                         STOP_ORBIT, OSV_LOCATION,
     &                         REF_FILETYPE, REFERENCE_FILENAME,
     &                         POF_FILETYPE,
     &                         OUTPUT_DIR, POF_FILENAME,
     &                         FILE_CLASS, VERSION_NUMBER, FH_SYSTEM,
     &                         IERR)
```

7.8.3 Input parameters

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

Table 31: Input parameters of xg_gen_pof 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. NOTE:Time correlations are only required if the input reference orbit file can not initialise them.	-	-
time_init	long*	-	Flag for selecting the time range of the initialisation.	-	Select either: · XO_SEL_ORBIT · XO_SEL_TIME
time_ref	long*	-	Time reference ID. (See note in the ref_filetype field)	-	Complete
start_time	double*	-	Processing time corresponding to the beginning of the required interval	Decimal days, MJD2000	[-18262.0,36524.0]
stop_time	double*	-	Processing time corresponding to the end of the required interval	Decimal days, MJD2000	[-18262.0,36524.0]
start_orbit	long*	-	Orbit number corresponding to the beginning of the required interval	orbits	>= 1
stop_orbit	long*	-	Orbit number corresponding to the end of the required interval	orbits	>= 1
osv_location	double*	-	Location of the state vector within the orbit	secs	>=0 < 1 nodal period
ref_filetype	long*	-	File type of the input reference file. (Note: When generating a POF file from a DORIS NAVIGATOR file, the input times should be expresed in UTC)	-	Complete
reference_filename	char*	-	Reference File name	-	
pof_filetype	long*	-	File type of the output reference file	-	XG_REF_FILETYPE_POF
output_dir	char*	-	Directory where the resulting POF is written (if NULL, the current directory is used)	-	-
pof_filename	char*	-	Output POF name if empty (i.e. “”), the software will generate the filename according to file name specification presented in [FORMATS]	-	-

Table 31: Input parameters of xg_gen_pof function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
file_class	char*	-	File class for output Predicted file	-	-
version_number	long*	-	Version number of output Predicted file	-	>= 0
fh_system	char*	-	System field of the output Predicted file fixed header	-	-

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 initialisation: time_init. See [ORBIT_SUM].
- Time reference: time_ref. See [ORBIT_SUM].
- File type: ref_filetype and pof_filetype. See this SUM.

7.8.4 Output parameters

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

Table 32: Output parameters of xg_gen_pof function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
ierr[XG_ERR_VECTOR_MAX_LENGTH]	long	all	Status vector	-	-

7.8.5 Warnings and errors

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

Table 33: Error messages of xg_gen_pof function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Wrong satellite flag	Computation not performed	XG_CFI_GEN_POF_WRONG_SAT_ID_ERR	0
ERR	Wrong input flag	Computation not performed	XG_CFI_GEN_POF_WRONG_FLAG_ERR	1
ERR	Time ID is not initialized	Computation not performed	XG_CFI_GEN_POF_TIME_INIT_ERR	2
ERR	Could not initialise the time reference	Computation not performed	XG_CFI_GEN_POF_TIME_INITIALIZATION_ERR	3
ERR	Cannot create in-memory XML tree	Computation not performed	XG_GEN_POF_CREATE_TREE_ERR	4
ERR	Cannot create root element	Computation not performed	XG_CFI_GEN_POF_CREATE_ROOT_ERR	5
ERR	Cannot add XML node to tree	Computation not performed	XG_CFI_GEN_POF_CREATE_NODE_ERR	6
ERR	Cannot initialise orbit	Computation not performed	XG_CFI_GEN_POF_ORBIT_INIT_FILE_ERR	7
ERR	Cannot initialise propagation	Computation not performed	XG_CFI_GEN_POF_PROPAG_INIT_ERR	8
ERR	Cannot initialise interpolation	Computation not performed	XG_CFI_GEN_POF_INTERPOL_INIT_ERR	9
ERR	Wrong interpol initialisation	Computation not performed	XG_CFI_GEN_POF_INTERNAL1_ERR	10
ERR	Cannot calculate state vector	Computation not performed	XG_CFI_GEN_POF_CALCULATING_STATE_VECTOR_ERR	11
ERR	Cannot convert time in processing reference	Computation not performed	XG_CFI_GEN_POF_TIME_ERR	12
ERR	Cannot convert time from processing to external	Computation not performed	XG_CFI_GEN_POF_TIME_TO_EXTERNAL_ERR	13
ERR	Cannot write state vector in XML file	Computation not performed	XG_CFI_GEN_POF_WRITE_SV_ERR	14

Table 33: Error messages of xg_gen_pof function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Error freeing memory	Computation not performed	XG_CFI_GEN_POF_CLOSE_ERR	15
ERR	Cannot write POF XML file	Computation not performed	XG_CFI_GEN_POF_FILE_ERR	16
ERR	Error creating the fixed header	Computation not performed	XG_CFI_GEN_POF_CREATE_HEADER_ERR	17

7.8.6 Runtime performances

The following runtime performance has been measured.

Table 34: Runtime performances of xg_gen_pof function

Ultra Sparc [ms]
TBD

7.8.7 Executable Program

The **gen_pof** executable program can be called from a Unix shell as:

gen_pof **-sat** satellite_name
-tref time_ref
{ **-tstart** start_time **-tstop** stop_time (decimal days) |
-tastart start_time **-tastop** stop_time (CCSDSA format) |
-ostart start_orbit **-ostop** stop_orbit (orbits) }
-osvloc osv_location (secs)
-reftyp ref_file_type
-ref reference_file
-poftyp pof_file_type
[**-dir** output_dir] (current directory by default)
[**-pof** output_filename] (default: name generated automatically)
[**-flcl** file_class] (empty string by default)
[**-vers** version] (version=0 by default)
[**-fhsys** fh_system] (empty string by default)
[**-v**]
[**-xl_v**]
[**-xo_v**]
[**-xp_v**]
[**-xg_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.
- [**-xl_v**] option for EXPLORER_LIB Verbose mode.
- [**-xo_v**] option for EXPLORER_ORBIT Verbose mode.
- [**-xp_v**] option for EXPLORER_POINTING Verbose mode.
- [**-xg_v**] option for EXPLORER_GEN_FILES 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 *satellite_name*: ERS1, ERS2, ENVISAT, METOP1, METOP2, METOP3, CRYOSAT, ADM, GOCE, SMOS.
- 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 *ref_file_type* and *pof_file_type*: OSF, POF, DORISNAV, ROF, DORISPREM, DORISPREC.
- Possible values for *time_ref* and *time_reference*: UNDEF, TAI, UTC, UT1, GPS.
- Time references need to be initialized only when using OSF as the type of the input reference file. The inputs needed for this issue are provided in the last three lines of parameters. 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)

Example:

```
gen_pof   -sat CRYOSAT -tref GPS -ostart 13 -ostop 14 -osvloc 0 -reftyp OSF
          -ref CS_TEST_MPL_ORBREF_20020301T122001_99999999T999999_0001.EEF
          -poftyp POF -dir ./gen_pof/ -pof orb_pre_file_at_307
          -tai -1100.1 -utc -1100.099595 -ut1 -1100.0995914352
          -gps -1100.0997801
```

7.9 xg_gen_dnf

7.9.1 Overview

The **xg_gen_dnf** CFI function creates a DORIS Navigator File using as input one of the following reference file types:

- Orbit Scenario File
- FOS Predicted Orbit File
- FOS Restituted Orbit File
- DORIS Navigator File
- DORIS Preliminary Orbit File
- DORIS Precise Orbit FileTime of the ascending crossing node (TAI, UTC and UT1)

The accepted output file types are:

- FOS Restituted Orbit File
- DORIS Preliminary Orbit File
- DORIS Precise Orbit FileTime

The time interval between consecutive OSVs can be selected by the user by means of a parameter in the calling interface. A flag for precise location of OSVs at “integer intervals” (e.g. every exact minute or every ten seconds) is also available. If the reference file and the DORIS Navigator File contain OSVs at the same time, these OSVs will be identical.
An optional control file can be introduced to correct the state vectors. This file contains the corrections for position and velocity in the along, across and radial directions. The format of this file is shown in the APPENDIX B. CONTROL FILE EXAMPLE.
Note: when using an OSF or Predicted Orbit file, the maximum time interval within the output Doris Navigator file is limited to 2 orbital periods before and after the middle point of the user requested time range.

7.9.2 Calling interface

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

```
#include <explorer_gen_files.h>
{
      long   sat_id;
      xl_time_id time_id;
      long   time_init, time_ref, start_orbit, stop_orbit,
             ref_filetype, dnf_filetype, osv_precise, version_number;
      double start_time, stop_time, osv_interval;
      char   reference_file[XG_MAX_LENGTH], output_dir[XG_MAX_LENGTH],
             dnf_filename[XG_MAX_LENGTH], ctrl_file[XG_MAX_LENGTH];
      char  *file_class, *fh_system;
      long   status, ierr[XG_ERR_VECTOR_MAX_LENGTH];
```

```
      status = long xg_gen_dnf(&sat_id, &time_id,
                               &time_init, &time_ref,
                               &start_time,&stop_time,
                               &start_orbit, &stop_orbit,
                               &osv_interval, &osv_precise,
                               &ref_filetype, reference_file, ctrl_file,
                               &dnf_filetype, output_dir, dnf_filename,
                               file_class, &version_number, fh_system,
                               /* output */
                               ierr);

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

      status = long xg_gen_dnf_run(&run_id,
                               &time_init, &time_ref,
                               &start_time,&stop_time,
                               &start_orbit, &stop_orbit,
                               &osv_interval, &osv_precise,
                               &ref_filetype, reference_file, ctrl_file,
                               &dnf_filetype, output_dir, dnf_filename,
                               file_class, &version_number, fh_system,
                               /* output */
                               ierr);
}
```

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

```
#include <explorer_gen_files.inc>

       INTEGER*4 SAT_ID, TIME_INIT, TIME_REF, START_ORBIT, STOP_ORBIT,
      & OSV_PRECISE, REF_FILETYPE, DNF_FILETYPE, VERSION_NUMBER
       REAL*8 START_TIME, STOP_TIME, OSV_INTERVAL
       CHAR*1 OUTPUT_DIR(XG_MAX_LENGTH), DNF_FILENAME(XG_MAX_LENGTH),
      & DNF_FILENAME(XG_MAX_LENGTH),REFERENCE_FILENAME(XG_MAX_LENGTH),
      & FILE_CLASS, FH_SYSTEM
      INTEGER*4 STATUS, IERR(XG_ERR_VECTOR_MAX_LENGTH)

      STATUS = XG_GEN_DNF( SAT_ID, TIME_INIT,
     &                     TIME_REF, START_TIME,
     &                     STOP_TIME, START_ORBIT,
```

```
     &                    STOP_ORBIT, OSV_INTERVAL, OSV_PRECISE
     &                    REF_FILETYPE, REFERENCE_FILENAME,
     &                    CTRL_FILE, DNF_FILETYPE,
     &                    OUTPUT_DIR, DNF_FILENAME,
     &                    FILE_CLASS, VERSION_NUMBER,FH_SYSTEM,
     &                    IERR)
```

7.9.3 Input parameters

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

Table 35: Input parameters of xg_gen_dnf 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. NOTE:Time correlations are only required if the input reference orbit file can not initialise them.	-	-
time_init	long*	-	Flag for selecting the time range of the initialisation.	-	Select either: · XO_SEL_ORBIT · XO_SEL_TIME
time_ref	long*	-	Time reference ID (see note in the ref_filetype field)	-	Complete
start_time	double*	-	Processing time corresponding to the beginning of the required interval	Decimal days, MJD2000	[-18262.0,36524.0]
stop_time	double*	-	Processing time corresponding to the end of the required interval	Decimal days, MJD2000	[-18262.0,36524.0]
start_orbit	long*	-	Orbit number corresponding to the beginning of the required interval	orbits	>= 1
stop_orbit	long*	-	Orbit number corresponding to the end of the required interval	orbits	>= 1

Table 35: Input parameters of xg_gen_dnf function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
osv_interval	double*	-	Interval between consecutive state vector. This parameter should be coherent with the osv_precise flag (see below). If osv_precise is set to: • XG_OSV_PRECISE_MINUTE: osv will be forced to be a multiple of 60 seconds. • XG_OSV_PRECISE_TEN_SECONDS: osv will be forced to be a multiple of 10 seconds.	secs	>=0
osv_precise	long*	-	Flag to indicate if state vectors should be placed at exact time locations	-	Complete
ref_filetype	long*	-	File type of the input reference file. (Note: When generating a DNF file from another DORIS NAVIGATOR file, the input times should be expresed in UTC)	-	Complete
reference_filename	char*	-	Reference File name	-	
ctrl_file	char*	-	Control File in xml format. This file contains the corrections for position and velocity in the along, across and radial directions together with the position accuracy(see APPENDIX B. CONTROL FILE EXAMPLE.) If empty string (“”), no corrections will be performed and the accuracy (quality index in the DNF)will be set to 1.	-	-
dnf_filetype	long*	-	File type of the output DORIS Navigator file	-	XG_REF_FILETYPE_DORIS_NAV
output_dir	char*	-	Directory where the resulting DNF is written (if NULL, the current directory is used)	-	-
dnf_filename	char*	-	Output DNF name if empty (i.e. “”), the software will generate the filename according to file name specification presented in [FORMATS]	-	-
file_class	char*	-	File class for output file (dummy in the current version)	-	-

Table 35: Input parameters of xg_gen_dnf function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
version_number	long*	-	Version number of output file (dummy in the current version)	-	>= 0
fh_system	char*	-	System field of the output file fixed header (dummy in the current version)	-	-

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 initialisation: time_init. See [ORBIT_SUM].
- Time reference: time_ref. See [ORBIT_SUM].
- OSV precise: osv_precise. See this SUM.
- File type: ref_filetype and rof_filetype. See this SUM.

7.9.4 Output parameters

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

Table 36: Output parameters of xg_gen_dnf function

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
ierr[XG_ERR_VECTOR_MAX_LENGTH]	long	all	Status vector	-	-

7.9.5 Warnings and errors

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

Table 37: Error messages of xg_gen_dnf function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Wrong satellite flag	Computation not performed	XG_CFI_GEN_DNF_WRONG_SAT_ID_ERR	0
ERR	Wrong input flag	Computation not performed	XG_CFI_GEN_DNF_WRONG_FLAG_ERR	1
ERR	Time ID is not initialized	Computation not performed	XG_CFI_GEN_DNF_TIME_INIT_ERR	2
ERR	Could not initialise the time reference	Computation not performed	XG_CFI_GEN_DNF_TIME_INITIALIZATION_ERR	3
ERR	Cannot create in-memory XML tree	Computation not performed	XG_CFI_GEN_DNF_CREATE_TREE_ERR	4
ERR	Cannot create root element	Computation not performed	XG_CFI_GEN_DNF_CREATE_ROOT_ERR	5
ERR	Cannot add XML node to tree	Computation not performed	XG_CFI_GEN_DNF_CREATE_NODE_ERR	6
ERR	Cannot initialise orbit ID	Computation not performed	XG_CFI_GEN_DNF_ORBIT_INIT_FILE_ERR	7
ERR	Cannot initialise the propagator	Computation not performed	XG_CFI_GEN_DNF_PROPAG_INIT_ERR	8
ERR	Could not perform a time <-> orbit transformation	Computation not performed	XG_CFI_GEN_DNF_TIME_ORBIT_ERR	9
ERR	Cannot initialise interpolation	Computation not performed	XG_CFI_GEN_DNF_INTERPOL_INIT_ERR	10
ERR	Wrong interpol initialisation	Computation not performed	XG_CFI_GEN_DNF_INTERNAL1_ERR	11
ERR	Cannot calculate state vector	Computation not performed	XG_CFI_GEN_DNF_CALCULATING_STATE_VECTOR_ERR	12
ERR	Cannot convert time to processing format	Computation not performed	XG_CFI_GEN_DNF_TIME_ERR	13
ERR	Cannot convert time from processing to external	Computation not performed	XG_CFI_GEN_DNF_TIME_TO_EXTERNAL_ERR	14

Table 37: Error messages of xg_gen_dnf function

Error type	Error message	Cause and impact	Error code	Error No
ERR	Cannot write state vector in XML file	Computation not performed	XG_CFI_GEN_DNF_WRITE_SV_ERR	15
ERR	Error freeing memory	Computation not performed	XG_CFI_GEN_DNF_CLOSE_ERR	16
ERR	Cannot write ROF XML file	Computation not performed	XG_CFI_GEN_DNF_FILE_ERR	17
ERR	Error creating the fixed header	Computation not performed	XG_CFI_GEN_DNF_CREATE_HEADER_ERR	18
ERR	Error opening the control file	Computation not performed	XG_CFI_GEN_DNF_CONTROL_FILE_OPEN_ERR	19
ERR	Control file does not cover the interval required	Computation not performed	XG_CFI_GEN_DNF_INVALID_CONTROL_FILE_ERR	20
ERR	Error reading the control file	Computation not performed	XG_CFI_GEN_DNF_CONTROL_FILE_READ_ERR	21
ERR	Error opening auxiliary file in the current directory	Computation not performed	XG_CFI_GEN_DNF_AUX_FILE_OPEN_ERR	22
ERR	Cannot correct state vector	Computation not performed	XG_CFI_GEN_DNF_CORRECT_OSV_ERR	23
ERR	Error writing ascii data in DORIS navigator file	Computation not performed	XG_CFI_GEN_DNF_ASCII_DATA_ERR	24
ERR	Error getting the parameter for the XML Header File	Computation not performed	XG_CFI_GEN_DNF_COMPUTE_HEADER_ERR	25
ERR	Error creating the XML Header File	Computation not performed	XG_CFI_GEN_DNF_CREATE_HEADER_FILE_ERR	26
ERR	OSV interval is not compatible with OSV Precise flag. The OSV Interval will be set to %f seconds.	Computation performed with a different value for the osv_interval	XG_CFI_GEN_DNF_WRONG_INTERVAL_WARN	27

7.9.6 Runtime performances

The following runtime performance has been measured.

Table 38: Runtime performances of xg_gen_dnf function

Ultra Sparc [ms]
TBD

7.9.7 Executable Program

The **gen_dnf** executable program can be called from a Unix shell as:

gen_dnf **-sat** satellite_name
-tref time_ref
{ -tstart start_time -tstop stop_time (decimal days) |
-tastart start_time -tastop stop_time (CCSDSA format) |
-ostart start_orbit -ostop stop_orbit (orbits) }
-osvint osv_interval
[**-osvpre**]
-reftyp ref_file_type
-ref reference_file
[**-ctrl** control_file]
[**-dir** output_dir] (current directory by default)
[**-dnf** output_filename] (default: name generated automatically)
[**-flcl** file_class] (empty string by default)
[**-vers** version] (version=0 by default)
[**-fhsys** fh_system] (empty string by default)
[**-v**]
[**-xl_v**]
[**-xo_v**]
[**-xp_v**]
[**-xg_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.
- [**-osvpre**] option for osv_precise.Default value is XG_OSV_PRECISE_NO. When the option is written, ovs_precise value is XG_OSV_PRECISE_MINUTE.
- [**-xl_v**] option for EXPLORER_LIB Verbose mode.
- [**-xo_v**] option for EXPLORER_ORBIT Verbose mode.
- [**-xp_v**] option for EXPLORER_POINTING Verbose mode.
- [**-xg_v**] option for EXPLORER_GEN_FILES 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 *satellite_name*: ERS1, ERS2, ENVISAT, METOP1, METOP2, METOP3, CRYOSAT, ADM, GOCE, SMOS.
- 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 *ref_file_type*: OSF, POF, DORISNAV, ROF, DORISPREM, DORISPREC.
- Possible values for *time_ref* and *time_reference*: UNDEF, TAI, UTC, UT1, GPS.
- Time references need to be initialized only when using OSF as the type of the input reference file. The inputs needed for this issue are provided in the last three lines of parameters. 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)

Example:

```
gen_dnf -sat CRYOSAT -tref UTC -tstart 0.99650462962963
        -tstop 01386574074708 -osvint 20 -reftyp ROF
        -ref EARTH_EXPLORER_FRO_TO_DORIS_2000
        -ctrl CONTROL_FILE.xml -dir ./gen_dnf/ -dnf doris_nav_at_308
        -tai 0.000000 -utc -4.0509259e-4 -ut1 -4.1435185185e-4 -gps 2.1991e-4
        -show
```

8 LIBRARY PRECAUTIONS

The following precautions shall be taken into account when using EXPLORER_GEN_FILES software library:

- When a message like

  EXPLORER_GEN_FILES >>> ERROR in *xg_function*: Internal computation error # *n*

  or

  EXPLORER_GEN_FILES >>> WARNING in *xg_function*: Internal computation warning # *n*

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

9 KNOWN PROBLEMS

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

Table 39: Known problems

CFI library	Problem	Work around solution
All	No Fortran version of the library exists	-

10 APPENDIX A. SWATH DEFINITION FILE EXAMPLE

Following there is an example of a Swath Definition File in XML format:

```
<?xml version = "1.0" encoding = "UTF-8"?>
<Earth_Explorer_File>
  <Earth_Explorer_Header>
    <Fixed_Header/>
    <Variable_Header>
      <Aocs_Amplitude>
        <Pitch unit="deg">-000.167200</Pitch>
        <Roll unit="deg">+000.050100</Roll>
        <Yaw unit="deg">+003.928400</Yaw>
      </Aocs_Amplitude>
    </Variable_Header>
  </Earth_Explorer_Header>
  <Data_Block type="xml">
    <List_of_Swath count="1">
      <Swath>
        <Output_File_Description>MERIS</Output_File_Description>
        <Output_File_Type>MERIS__501</Output_File_Type>
        <Swath_Type>line</Swath_Type>
        <Num_Swath_Records>1200</Num_Swath_Records>
        <Refraction>
          <Model>NO_REF</Model>
          <Freq unit="MHz">000440000000</Freq>
        </Refraction>
        <Line_Geometry>
          <Left_Pt>
            <Azimuth unit="deg">+270.000000</Azimuth>
            <Elevation unit="deg">+055.750000</Elevation>
            <Altitude unit="m">+000000.000</Altitude>
          </Left_Pt>
          <Mid_Pt>
            <Azimuth unit="deg">+090.000000</Azimuth>
            <Elevation unit="deg">+090.000000</Elevation>
            <Altitude unit="m">+000000.000</Altitude>
          </Mid_Pt>
          <Right_Pt>
            <Azimuth unit="deg">+090.000000</Azimuth>
            <Elevation unit="deg">+055.750000</Elevation>
            <Altitude unit="m">+000000.000</Altitude>
          </Right_Pt>
        </Line_Geometry>
        <No_Mispointing/>
        <No_Instrument_Dependent/>
      </Swath>
    </List_of_Swath>
  </Data_Block>
</Earth_Explorer_File>
```

11 APPENDIX B. CONTROL FILE EXAMPLE

Following there is an example of a Control File in XML format:

```
<?xml version="1.0" ?>
<Earth_Explorer_File>
    <Earth_Explorer_Header>
        </Fixed_Header>
        <Variable_Header>
            <START_TIME>01-JAN-2000 23:00:00.000000</START_TIME>
            <STOP_TIME>02-JAN-2000 23:00:00.000000</STOP_TIME>
            <NUM_SOURCE_PACKETS>+8640</NUM_SOURCE_PACKETS>
        </Variable_Header>
    </Earth_Explorer_Header>

    <Data_Block type="xml">
        <List_of_num_nsp_rec count="8640">
            <nsp_rec>
                 <TIME unit="s">+00000</TIME>
                 <delta_pos>
                    <RADIAL unit="m">+00000.000</RADIAL>
                    <ACROSS unit="m">+00000.000</ACROSS>
                    <ALONG unit="m">+00000.000</ALONG>
                 </delta_pos>
                 <delta_vel>
                    <RADIAL unit="m/s">+00.000000</RADIAL>
                    <ACROSS unit="m/s">+00.000000</ACROSS>
                    <ALONG unit="m/s">+00.000000</ALONG>
                 </delta_vel>
                 <accuracy>
                    <POSITION unit="m">000.0</POSITION>
                 </accuracy>
            </nsp_rec>
            <nsp_rec>
                 <TIME unit="s">+00010</TIME>
                 <delta_pos>
                    <RADIAL unit="m">+00000.000</RADIAL>
                    <ACROSS unit="m">+00000.000</ACROSS>
                    <ALONG unit="m">+00000.000</ALONG>
                 </delta_pos>
                 <delta_vel>
                    <RADIAL unit="m/s">+00.000000</RADIAL>
                    <ACROSS unit="m/s">+00.000000</ACROSS>
                    <ALONG unit="m/s">+00.000000</ALONG>
                 </delta_vel>
                 <accuracy>
                    <POSITION unit="m">000.0</POSITION>
                 </accuracy>
            </nsp_rec>
        </List_of_num_nsp_rec>
```

```
    </Data_Block>
</Earth_Explorer_File>
```