RE-ENGINEERING OF MISSION ANALYSIS SOFTWARE FOR ENVISAT-1

PPF_GENREF SOFTWARE USER MANUAL

PO-IS-DMS-GS-0609

Code:	PO-IS-DMS-GS-0609
Issue:	3.9
Date:	30/05/11

	Name	**Function**	**Signature**
Prepared by:	M. Sánchez-Gestido (GMV) Carlos Villanueva (Deimos-Space) Juan Jose Borrego (Deimos-Space)	Project Engineer	
Checked by:	José Antonio González Abeytua	Project Manager	
Approved by:	José Antonio González Abeytua	Project Manager	

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

Document Information

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

Internal Distribution		
Name	Unit	Copies

External Distribution		
Name	Organisation	Copies

Archiving	
Word Processor:	Framemaker 6.0
File name:	envcfi-dms-sum-006-10
Archive Code	P/SUM/DMS/01/043-023

Document Status Log

Issue	Change Description	Date	Approval
1.0	•	11/04/97	
2.0	• It includes the modifications with respect the prototype functions	20/10/97	
2.1	• It includes the description of algorithms	05/12/97	
2.2	• Runtime performances improved • Known problems of release 2.1 solved	18/05/98	
2.3	• Error handling vectors length modified	19/10/98	
2.4	• Cosmetic changes. It points to new versions of the other libraries.	31/05/99	
2.5	• New release.It points to new versions of the other libraries.	14/04/00	
2.6	• Unreleased document	22/06/01	
2.7	• Unreleased document	31/07/01	
2.8	• Unreleased document	22/10/01	
2.9	• Unreleased document	18/01/02	
3.1	• Unreleased	25/11/02	
3.2	• New WINDOWS version. • Document format change.	26/05/03	
3.3	• Maintenance Release	13/12/04	
3.3.1	• Maintenance Release	15/02/05	
3.4	• Maintenance Release	17/05/05	
3.5	• Maintenance Release • Library Optimization	30/01/06	
3.6	• Maintenance Release • New library for LINUX 64-bits	22/12/06	
3.7	• Changes for Envisat extended life • New library for MAC OS on Intel platforms • Maintenance Release	11/04/08	

Issue	Change Description	Date	Approval
3.8	• Changes for Envisat extended life • New function pg_genosf_ext_cyc • New function pg_genswath_ext_cycle • Maintenance release	15/09/09	
3.9	• Maintenance release.	30/05/11	

Table of Contents

1 SCOPE

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

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

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

2 ACRONYMS AND NOMENCLATURE

2.1 Acronyms

AOCS	Attitude and Orbit Control System
ANX	Ascending Node Crossing
CFI	Customer Furnished Item
CS	Coordinate System
DRS	Data Relay Satellite
ESA	European Space Agency
ESTEC	European Space Technology and Research Centre
FOS	Flight Operation Segment
GS	Ground Station
H/W	Hardware
I/F	Interface
LOS	Line Of Sight
PPF	Polar Platform
RAM	Random Access Memory
SBT	Satellite Binary Time
SOBM	Sun Occultation By Moon
SUM	Software User Manual
S/W	Software
SZA	Sun Zenith Angle
UTC	Universal Time Coordinated
UT1	Universal Time UT1
SSP	Sub Satellite Point

2.2 Nomenclature

CFI A group of CFI functions, and related software and documentation, that will be distributed by ESA to the users as an independent unit

CFI function A single function within a CFI that can be called by the user

Library A software library containing all the CFI functions included within a CFI plus the supporting functions used by those CFI functions (transparently to the user)

3 APPLICABLE AND REFERENCE DOCUMENTS

3.1 Applicable documents

AD 1 Re-engineering of Mission Analysis Software for Envisat-1: Statement of Work.PO-SW-ESA-GS-0344. ESA/ESTEC/NW. Issue 3.0. 19/12/1995.

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

3.2 Reference documents

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

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

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

RD 4 Envisat-1 Reference Operation Plan (ROP). EN-PL-ESA-GS-00334

RD 5 Envisat-1 Mission CFI Software. PPF_Visibility Software User Manual. PO-IS-DMS-GS-0560

4 INTRODUCTION

This software package contains the CFI functions required to compute the reference orbit event file, used for ENVISAT-1 mission planning purposes, and the instrument swath template files, used for instance by the time segment calculation CFI functions of PPF_VISIBILITY.

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 of Envisat-1 operations.

The PPF_GENREF library includes the following CFI functions:

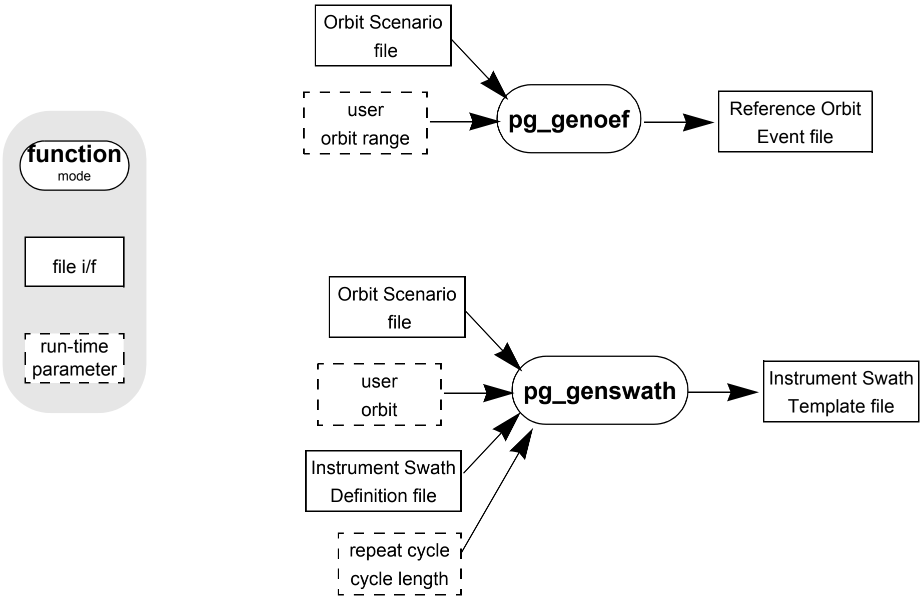
- **pg_genoef**: generates the reference orbit event file for a range of orbits
- **pg_genswath**: generates the instrument swath template file for a given instrument mode and orbit
- **pg_genswath_ext_cycle**: generates the instrument swath template file for a given instrument mode and orbit with a base of an orbital change for every cycle
- **pg_genosf_ext_cycle**: function generates an orbit scenario file containg a new orbital change for every cycle.

Several files are required to operate properly the above functions:

- Orbit Scenario File (**all functions**)
- Instrument Swath Definition Files (**pg_genswath**)

An overview of the PPF_GENREF data flow is presented in Figure 1:

Figure 1: PPF_GENREF data flow



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

Please refer also to:

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

5 LIBRARY INSTALLATION

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

6 LIBRARY USAGE

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

- `PPF_LIB` (version 5.8, see RD 3).
- `PPF_ORBIT` (version 5.8, see RD 3).
- `PPF_POINTING` (version 5.8, see RD 3).

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

- `ppf_genref.h` (for a C application)
- `ppf_genref.inc` (for a Fortran application)

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 RD 3 for installation procedures):

- SOLARIS / AIX

```
-Icfi_include_dir -Lcfi_lib_dir
-lppf_genref
-lppf_pointing -lppf_orbit -lppf_lib
```

- WINDOWS

```
/I "cfi_include_dir" /libpath:"cfi_lib_dir" libppf_genref.lib
libppf_pointing.lib libppf_orbit.lib libppf_lib.lib
```

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

To avoid problems in linking a user application with the PPF_GENREF 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 `PG_` or `pg_`.

To preserve compatibility with the historical CFI function names, it is possible to call the CFI functions described in this document from a user application with or without the `pg_` prefix.

This is summarized in the table below.

Function Name	Enumeration value	long
Main CFI Functions		
pg_genref genref	PG_GENOEF_ID	0
pg_genswath genswath	PG_GENSWATH_ID	1
Error Handling Functions		

Function Name	Enumeration value	long
pg_verbose	not applicable	
pg_silent		
pg_vector_code		
pg_vector_msg		
pg_print_msg		

Notes about the table:

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

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 pg_genoef

7.1.1 Overview

The **pg_genoef** function generates the Reference Orbit Event File. The format of this file is described in RD 4. This file lists for every orbit the following parameters:

- Absolute orbit number
- Relative orbit number, cycle number and phase number
- Ascending node time, crossing the Earth-fixed Equator from South to North, in UTC
- Satellite repeat cycle and cycle length, in days respectively in orbits
- Ascending node longitude, in degrees
- Mean local solar time, in hours
- Mean Kepler state vector
- Cartesian state vector, in meters respectively meters / second
- Satellite eclipse exit time and eclipse entry time, in seconds since ascending node
- Epoch when nadir Sun zenith angle (SZA) = 90 degree, in seconds since ascending node
- Any other SZA times (the number of SZA angles in OSF is limited to 10).

Also for the complete orbit range, the Reference Orbit Event File contains

- Epoch of Sun eclipse by Moon, Orbit number and seconds since ascending node

The input of **pg_genoef** is the Orbit Scenario File. This input file describes the first orbit of the mission (starting at first ascending node after satellite injection) and any other satellite orbit where the orbit characteristics are changing (=repeat cycle or phasing change), with a given repeat cycle and cycle length, longitude of ascending node, and a given Mean Local Solar Time or UTC.

The CFI function **pg_genoef** is capable of calculating an Orbit Event File for sun-synchronous orbits with a MLST at ascending node between 20:30 and 23:00.

The CFI function **pg_genoef** verifies the Orbit Scenario File by checking the following conditions.

- Consistency of absolute orbit numbering (shall be strict monotonic increasing)
- Consistency of Relative orbit numbering:
    - relative orbits increment every orbit at ascending node (if number larger than current cycle length then reset to 1)
    - relative orbit 1 has a longitude between 0.0 inclusive and 360.0/(CycleLength) East exclusive. Relative orbit N has a longitude between ((1-N)*360*RepeatCycle/CycleLength) inclusive and ((1-N)*360*RepeatCycle/CycleLength + 360.0/CycleLength) exclusive
    - cycle numbers increment when relative number is reset to 1 or when a new repeat cycle / cycle length is requested (then relative orbit can have a discontinuity)
- Phase only increments by one as requested by the Orbit Scenario File
- Consistency of given UTC and UTC based on longitude and MLST (shall be smaller than 1 sec). **pg_genoef** will actually use the UTC based on longitude and MLST.
- Consistency of given Date (time-jump within 12 sec + 40 sec for respectively change in longitude and change in MLST)
- Consistency of given Longitude (the longitude of the ascending node should not jump more than 0.05 deg)

The Reference Orbit Event File also contains the orbital changes information extracted from the input Orbit Scenario File.

7.1.2 Calling sequence pg_genoef

For C programs, the call to **pg_genoef** is (input parameters are underlined):

```
#include ”ppf_genref.h”
#define MAX_SZA <your value here>
{
      long     start_orbit, stop_orbit, ierr[10], status;
      char     *orbit_scenario_file,*reference_orbit_event_file;

      status = pg_genoef (  orbit_scenario_file,
                            &start_orbit, &stop_orbit,
                            reference_orbit_event_file, ierr);

/* test status */
```

For FORTRAN programs **pg_genoef** has the following calling sequence (input parameters are underlined, note that the C preprocessor must be used because of the presence of the #include statement):

```
#include ”ppf_genref.inc”

       INTEGER*4   START_ORBIT, STOP_ORBIT, IERR(10), STATUS
       CHARACTER*(*)    ORBIT_SCENARIO_FILE, REFERENCE_ORBIT_EVENT_FILE


       STATUS = PG_GENOEF ( ORBIT_SCENARIO_FILE,
      &                     START_ORBIT, STOP_ORBIT,
      &                     REFERENCE_ORBIT_EVENT_FILE, IERR)

C test status
```

7.1.3 Input parameters pg_genoef

c name	c type	Array Element	Description	Units	Range
orbit_scenario_file	char *		Orbit Scenario File, defining the Envisat-1 reference orbit, by describing the orbits with a changing orbit characteristic. Defines start_osf and stop_osf, the first and last orbit of the header of the current file.		
start_orbit	long		First orbit Reference Orbit Event File. The Reference Orbit Event File will be generated from the beginning of first orbit (within orbit range from orbit_scenario_file) If set to zero then first orbit of orbit_scenario_file is selected.	absolute orbit number	= 0 or >=start_osf <=stop_osf
stop_orbit	long		Last orbit Reference Orbit Event File. The Reference Orbit Event File will be generated until the end of last orbit (within orbit range from orbit_scenario_file) If set to zero then last orbit of orbit_scenario_file is selected.	absolute orbit number	= 0 or >= start_orbit <= stop_osf
orbit_event_file	char *		name of output Reference Orbit Event File If empty (i.e. ""), the software will generate the name according to file name specification presented in RD 4		

Table 1: Input parameters for pg_genoef

7.1.4 Output parameters pg_genoef

c name /	c type /	Array Ele-ment	Description	Range
pg_genoef	long		Function status flag, 0 No error > 0 Warnings, results generated < 0 Error, no results generated	
ierr[10]	long	all	Error status flags	

Table 2: Output parameters for pg_genoef

7.1.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **pg_genoef** CFI function after translating the returned status vector into the equivalent list of error messages by calling the function of the PPF_GENREF software library **pg_vector_msg** (see RD 3).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation, 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 **pg_genref** CFI function by calling the function of the PPF_GENREF software library **pg_vector_code** (see RD 3).

Error type	Error message	Cause and impact	Error Code	Error No
ERR	Error in Orbit Scenario File definition (number of Sun Zenith Angles, number of Orbit Changes, incomplete Orbit Scenario File record, ...). [PG]	Computation not perfomed	PG_CFI_GENOEF_OSF_WRONG_ERR	0
ERR	Auxiliary file to write Orbit Scenario File information cannot be opened.	Computation not perfomed	PG_CFI_GENOEF_OSF_INFO_OPEN_ERR	1
ERR	Orbit Scenario File information cannot be written in auxiliary file.	Computation not perfomed	PG_CFI_GENOEF_OSF_INFO_WRITE_ERR	2
ERR	Orbit Scenario File cannot be opened.	Computation not perfomed	PG_CFI_GENOEF_OSF_OPEN_ERR	3
ERR	Error while opening auxiliary file to write Sun Occultation by Moon information.	Computation not perfomed	PG_CFI_GENOEF_SOBM_OPEN_ERR	4
ERR	Error while opening Orbit Event File to write.	Computation not perfomed	PG_CFI_GENOEF_OEF_OPEN_ERR	5
ERR	Cannot read Orbit Scenario File record: (%ld). [PG]	Computation not perfomed	PG_CFI_GENOEF_PG_OSF_REC_READ_ERR	6
ERR	Absolute orbit numbering must be strictly monotonic: Orbit Scenario File record: (%ld).	Computation not perfomed	PG_CFI_GENOEF_ABS_ORB_COUNT_ERR	7
ERR	Error in some orbit change paramaters: Orbit Scenario File record: (%ld). [PG]	Computation not perfomed	PG_CFI_GENOEF_PG_ORB_CHANG_ERR	8
ERR	Sun Occultation by Moon entry without Exit not followed by Exit without Entry. Orbit: (%ld).	Computation not perfomed	PG_CFI_GENOEF_SOBM_ENTR_EXIT_ERR	9

Error type	Error message	Cause and impact	Error Code	Error No
ERR	Error writing Sun Occultation by Moon record. Orbit: (%ld).	Computation not perfomed	PG_CFI_GENOEF_SOBM_WRITE_ERR	10
WARN	There is an exit without an entry in the 1st Orbit Event File orbit. Orbit: (%ld).	Computation perfomed Message to inform the user	PG_CFI_GENOEF_SOBM_1ST_ORB_WARN	11
ERR	There is an exit without an entry in the previous orbit. Orbit: (%ld).	Computation not perfomed	PG_CFI_GENOEF_SOBM_1ST_ORB_ERR	12
ERR	Error writing Orbit Event File record. Orbit: (%ld).	Computation not perfomed	PG_CFI_GENOEF_PG_OEF_REC_WRITE_ERR	13
ERR	Error while opening auxiliary file to write Variable Header information.	Computation not perfomed	PG_CFI_GENOEF_VHR_OPEN_ERR	14
ERR	Error while writing beginning of Orbit Event File.	Computation not perfomed	PG_CFI_GENOEF_OEF_BEG_WRITE_ERR	15
ERR	Error writing Orbit Event File variable header.	Computation not perfomed	PG_CFI_GENOEF_PG_OEF_VHR_WRITE_ERR	16
ERR	Error while writing end of Sun Occultation by Moon list.	Computation not perfomed	PG_CFI_GENOEF_SOBM_END_WRITE_ERR	17
ERR	Error while writing end of Orbit Scenario File information in Orbit Event File.	Computation not perfomed	PG_CFI_GENOEF_OEF_END_WRITE_ERR	18
ERR	Error in Orbit Scenario File definition (Fixed Header, Variable Header or two first used Orbit Event File records in Orbit Scenario File). [PG].	Computation not perfomed	PG_CFI_GENOEF_PG_OSF_CHECK_ERR	32
ERR	Error in Orbit Scenario File definition, orbit change: (%ld). [PG]	Computation not perfomed	PG_CFI_GENOEF_PG_OSF_CHECK_NTH_ERR	33
ERR	Error while copying Orbit record. Orbit: (%ld).	Computation not perfomed	PG_CFI_GENOEF_PG_ORB_REC_COPY_ERR	34
ERR	Error while computing Ascending Node parameters. Orbit: (%ld).	Computation not perfomed	PG_CFI_GENOEF_PG_GENSTATE_ERR	35
ERR	Error computing eclipse times. Orbit: (%ld).	Computation not perfomed	PG_CFI_GENOEF_PG_ECLIPSE_ERR	36
ERR	Error computing Sun Zenith Angles ocurrence times. Orbit: (%ld).	Computation not perfomed	PG_CFI_GENOEF_PG_SUN_ZEN_ANGL_ERR	37
ERR	Sun Zenith Angle: (%.2lf) is greater than maximum that can be computed for this orbit.	Computation not perfomed	PG_CFI_GENOEF_PG_SUN_ZEN_ANGL_SZA_MAX_ERR	38

Error type	Error message	Cause and impact	Error Code	Error No
ERR	Sun Zenith Angle: (%.2lf) is smaller than minimum that can be computed for this orbit.	Computation not perfomed	PG_CFI_GENOEF_PG_SUN_ZEN_ANGL_SZA_MIN_ERR	39
ERR	Error computing Sun Occultation by Moon. Orbit: (%ld).	Computation not perfomed	PG_CFI_GENOEF_PG_SUN_OCC_BY_MOON_ERR	40
WARN	Sun Occultation by Moon found in first or last orbit searched. Orbit: (%ld).	Computation perfomed Message to inform the user	PG_CFI_GENOEF_PG_SUN_OCC_BY_MOON_WARN	41
ERR	Error in Orbit Event File name generation.	Computation not perfomed	PG_CFI_GENOEF_PG_FILENAME_GENER_ERR	42
ERR	Error writing Orbit Event File fixed header.	Computation not perfomed	PG_CFI_GENOEF_PG_FHR_WRITE_ERR	43
ERR	Error concatenating files.	Computation not perfomed	PG_CFI_GENOEF_CAT_FILES_ERR	44
ERR	Error removing files.	Computation not perfomed	PG_CFI_GENOEF_REMOVE_FILES_ERR	45
WARN	(Entry/no Exit) in last orbit.	Computation perfomed Message to inform the user	PG_CFI_GENOEF_SOBM_LAST_ORBIT_WARN	46
ERR	Sun Occultation by Moon entry found in previous orbits but when the end of the file is reached there is no (Exit/no Entry) in last orbit.	Computation not perfomed	PG_CFI_GENOEF_SOBM_LAST_ORBIT_ERR	47

7.1.6 Runtime performances

The following runtime performances have been measured:

Orbits	SZA	Ultra Sparc [sec]
100	2	27.0
501	2	94.5
1800	2	313.0

7.1.7 Executable Program genoef

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

```
genoef -osf orbit_scenario_file_name
      [ -oef orbit_event_file_name ]
        -start start_orbit_number
        -stop stop_orbit_number
      [ -v ]
      [ -pl_v ]
      [ -po_v ]
      [ -pp_v ]
      [ -pg_v ]
      [ -h or -help ]
```

Note that:

- order of parameters does not matter
- bracketed parameters are not mandatory (check previous paragraphs for behaviour when not provided)
- -pl_v for PPF_LIB verbose mode
- -po_v for PPF_ORBIT verbose mode
- -pp_v for PPF_POINTING verbose mode
- -pg_v for PPF_GENREF verbose mode
- -v for Verbose mode for the four libraries (default is Silent)
- -h or -help option will result in printing the above text on stderr (no execution)
- all errors during execution will be printed in clear messages on stderr

7.2 pg_genswath

7.2.1 Overview

The **pg_genswath** 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 **ppf_visibility**.

The **pg_genswath** function contains for each instrument swath type a swath calculation algorithm. The selection of the algorithm and its corresponding parameters are defined in the instrument swath definition file. One instrument swath definition file can contain several instrument swath definitions. The swath definition file is described in Appendix A.

The instrument swath template file, consists of an ASCII header, listing the instrument mode, the swath type, the repeat cycle and cycle length, and referencing to the swath definition file. The header also 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 3 points[1]. These points are located from left to right when looking in the flight direction (e.g. for ASAR: from near-swath, via mid-swath, to far-swath). For a drawing describing the swath configuration, please refer to Figure 2 in RD 5.

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

[1] For line swath, e.g. the RA-2 swath, only one point is given for each time.

7.2.2 Swath Definition

Described in section **7.1.2 Swath definition** in RD 5.

7.2.2.1 Earth-observing Instruments Swath Definition

Described in section **7.1.2.1 Earth-observing Instruments Swath Definition** in RD 5.

7.2.2.2 Limb-sounding Instruments Swath Definition

Described in section **7.1.2.2 Limb-sounding Instruments Swath Definition** in RD 5.

7.2.2.3 Limb-sounding Instruments Inertial Swath Definition

Both Gomos occultation mode and Mipas Line of Sight mode, observe inertial targets. For the CFI function **pv_starvistime** (availability TBC) the FOV direction in inertial coordinates must be available. Therefore for these instrument modes the direction in inertial space, for a given tangent altitude, is given in the swath template file.

7.2.2.4 Instrument Swath Definition File

Format is described in Appendix A. Each swath calculation mode has its own set of parameters, so the format of the file is calculation mode dependent.

7.2.3 Calling sequence pg_genswath

For C programs, the call to **pg_genswath** is (input parameters are underlined):

```
#include ”ppf_genref.h”
{
       long            requested_orbit ;
       long            repeat_cycle, cycle_length ;
       long            status, ierr[10];
       char            *orbit_scenario_file, *swath_definition_file ;
       char            *instr_swath_file_suffix ;

       status = pg_genswath (
                        orbit_scenario_file, &requested_orbit,
                        swath_definition_file, instr_swath_file_suffix,
                        &repeat_cycle,&cycle_length, ierr);

/* test status */
}
```

For FORTRAN programs **pg_genswath** has the following calling sequence (input parameters are underlined, note that the C preprocessor must be used because of the presence of the #include statement):

```
#include ”ppf_genswath.inc”
        INTEGER*4   REQUESTED_ORBIT
        INTEGER*4   FIRST_ORBIT, LAST_ORBIT, REPEAT_CYCLE, CYCLE_LENGTH
        INTEGER*4   STATUS, IERR(10)
        CHARACTER*(*)    ORBIT_SCENARIO_FILE, SWATH_DEFINITON_FILE
        CHARACTER*(*)    INSTR_SWATH_FILE_SUFFIX

        STATUS = PG_GENSWATH (
       &                ORBIT_SCENARIO_FILE, REQUESTED_ORBIT,
       &                SWATH_DEFINITON_FILE, INSTR_SWATH_FILE_SUFFIX,
       &                REPEAT_CYCLE, CYCLE_LENGTH, IERR)


C test status
```

7.2.4 Input parameters pg_genswath

c name	c type	Description	Units	Range
orbit_scenario_file	char *	Orbit Scenario File, defining the Envisat-1 reference orbit, by describing the orbits with a changing orbit characteristic.		
requested_orbit	long	Orbit for which the instrument swath template file will be calculated. If non zero, then repeat cycle and cycle length of the orbit are looked up in the orbit_scenario_file If set to zero, then use directly repeat_cycle and cycle_length parameters (see below)	absolute orbit number	= 0 or >= start_osf <= stop_osf
swath_definition_file	char *	File name of the instrument swath definition file. Format see section 7.2.2.4.		
instrument_swath_file_suffix	char *	name suffix for output swath file. Prefix if deterministic, based on swath name (6 characters, see Table A.3) and cycle length (4 characters, leading underscores "_" is needed): Example: ASARGS_501 If empty (i.e. ""), the software will generate the suffix according to file name specification presented in RD 4		
Repeat Cycle	long	Repeat Cycle If requested_orbit is not set to zero, use requested_orbit and Orbit Scenario File (see above) to determine repeat cycle AND cycle length	days	
Cycle Length	long	Cycle Length of requested_orbit If requested_orbit is not set to zero, use requested_orbit and Orbit Scenario File (see above) to determine repeat cycle AND cycle length	orbits	

Table 3: Input parameters for pg_genswath

7.2.5 Output parameters pg_genswath

c name	c type	Array Ele-ment	Description	Unit	Range
pg_genswath	long		Function status flag, 0 = No error > 0 Warnings, results generated < 0 Error, no results generated		
ierr[10]	long		Error status flags		

Table 4: Output parameters for pg_genswath

7.2.6 Warnings and errors

Next table lists the possible error messages that can be returned by the **pg_genswath** CFI function after translating the returned status vector into the equivalent list of error messages by calling the function of the PPF_GENREF software library **pg_vector_msg** (see RD 3).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation, 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 **pg_genref** CFI function by calling the function of the PPF_GENREF software library **pg_vector_code** (see RD 3).

Error type	Error message	Cause and impact	Error Code	Error No
ERR	Cannot open Swath Definition File	Computation not performed	PG_CFI_GENSWATH_OPEN_SDF_ERR	0
ERR	Unable to store Swath Template File names (not enough memory).	Computation not perfomed	PG_CFI_GENSWATH_STF_FILENAMES_ERR	1
ERR	Error in "SWATH=". There is no cycle length value. Swath Definition File record: %ld.	Computation not perfomed	PG_CFI_GENSWATH_SWATH_ID_CYCL_LENGTH_1ST_ERR	2
ERR	Cycle length value in "SWATH=" does not match with cycle length for computations. Swath Definition File record: %ld	Computation not perfomed	PG_CFI_GENSWATH_SWATH_ID_CYCL_LENGTH_2ND_ERR	3
ERR	Cannot open Swath Template File for writting. Swath Template File: %ld failed.	Computation not perfomed	PG_CFI_GENSWATH_OPEN_STF_WRITE_ERR	4
ERR	Error while writing beginning of Swath Template File. Swath Template File: %ld failed.	Computation not perfomed	PG_CFI_GENSWATH_STF_BEG_WRITE_ERR	5
ERR	Error while writing comment line. Swath Template File: %ld failed.	Computation not perfomed	PG_CFI_GENSWATH_COMM_LINE_ERR	6
ERR	Cannot rename Swath Template File: %ld.	Computation not perfomed	PG_CFI_GENSWATH_RENAME_FILES_ERR	7
ERR	Error in Genswath inputs (REPEAT CYCLE, CYCLE LENGTH or Orbit Scenario File inconsistency).	Computation not perfomed	PG_CFI_GENSWATH_PG_CHECK_GENSW_INP_ERR	32
ERR	Error while generating Ascending Node parameters.	Computation not perfomed	PG_CFI_GENSWATH_PG_GENSTATE_ERR	33
ERR	Error in Genswath inputs. [PG]	Computation not perfomed	PG_CFI_GENSWATH_PG_SDF_PREV_CHECK_ERR	34

Error type	Error message	Cause and impact	Error Code	Error No
ERR	"SWATH=" value of Swath record: %ld is repeated below.	Computation not perfomed	PG_CFI_GENSWATH_PG_SDF_PREV_CHECK_SWATH_VALUE_ERR	35
ERR	Error reading Swath record: %ld.	Computation not perfomed	PG_CFI_GENSWATH_PG_SDF_PREV_CHECK_SWATH_RECORD_ERR	36
ERR	Error reading Swath Definition File header.	Computation not perfomed	PG_CFI_GENSWATH_PG_SDF_READ_HEADER_ERR	37
ERR	Error reading Swath Definition File record: %ld. [PG]	Computation not perfomed	PG_CFI_GENSWATH_PG_SDF_REC_READ_ERR	38
ERR	Error in algorithm computations. Swath Template File: %ld failed.	Computation not perfomed	PG_CFI_GENSWATH_PG_ALGOR_ERR	39
ERR	Error while generating output filename. Swath Template File: %ld failed.	Computation not perfomed	PG_CFI_GENSWATH_PG_FILENAME_GENER_SWATH_ERR	40
ERR	Error writting fixed header. Swath Template File: %ld failed.	Computation not perfomed	PG_CFI_GENSWATH_PG_FHR_WRITE_ERR	41
ERR	Error while writing Swath Template File variable header. Swath Template File: %ld failed.	Computation not perfomed	PG_CFI_GENSWATH_PG_STF_VHR_WRITE_ERR	42

7.2.7 Runtime performances

The following runtime performances have been measured:

Swath Algorithm (Instrument)	Points	Ultra Sparc [sec]
Line (`MERIS_`)	1200	13.0
Limb (`MIP_RH`)	1200	18.0
Point (`RA_2__`)	1200	6.0
ASAR (`SA1IM_`)	1200	20.0
Inertial (`MIPIRH`)	1200	12.0

7.2.8 Executable Program genswath

the **genswath** executable program can be called from a Unix shell as:

```
genswath [-osf orbit_scenario_file_name -orbit orbit_number ]
         [ -recyc repeat_cycle -cyclen cycle_length ]
           -sdf swath_definition_file_name
         [ -isfx instrument_swath_file_name_suffix ]
         [ -v ]
         [ -pl_v ]
         [ -po_v ]
         [ -pp_v ]
         [ -pg_v ]
         [ -h or -help ]
```

Note that:

- order of parameters does not matter
- bracketed parameters are not mandatory (check previous paragraphs for behaviour when not provided)
- 1st 2 lines of parameters are mutually exclusive
- -pl_v for PPF_LIB verbose mode
- -po_v for PPF_ORBIT verbose mode
- -pp_v for PPF_POINTING verbose mode
- -pg_v for PPF_GENREF verbose mode
- -v for Verbose mode for the four libraries (default is Silent)
- -h or -help option will result in printing the above text on stderr (no execution)
- all errors during execution will be printed in clear messages on stderr

7.3 pg_genswath_ext_cycle

7.3.1 Overview

The **pg_genswath_ext_cycle** function generates for the different instrument modes the corresponding instrument swath template file for input files of version 2. These template files define the swaths to be used in the segment calculation routines of **ppf_visibility**.

The **pg_genswath_ext_cycle** function contains for each instrument swath type a swath calculation algorithm. The selection of the algorithm and its corresponding parameters are defined in the instrument swath definition file. One instrument swath definition file can contain several instrument swath definitions. The swath definition file is described in Appendix A.

The instrument swath template file, consists of an ASCII header, listing the instrument mode, the swath type, the repeat cycle and cycle length, and referencing to the swath definition file. The header also 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 3 points[2]. These points are located from left to right when looking in the flight direction (e.g. for ASAR: from near-swath, via mid-swath, to far-swath). For a drawing describing the swath configuration, please refer to Figure 2 in RD 5.

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

[2] For line swath, e.g. the RA-2 swath, only one point is given for each time.

7.3.2 Swath Definition

Described in section **7.1.2 Swath definition** in RD 5.

7.3.2.1 Earth-observing Instruments Swath Definition

Described in section **7.1.2.1 Earth-observing Instruments Swath Definition** in RD 5.

7.3.2.2 Limb-sounding Instruments Swath Definition

Described in section **7.1.2.2 Limb-sounding Instruments Swath Definition** in RD 5.

7.3.2.3 Limb-sounding Instruments Inertial Swath Definition

Both Gomos occultation mode and Mipas Line of Sight mode, observe inertial targets. For the CFI function **pv_starvistime** (availability TBC) the FOV direction in inertial coordinates must be available. Therefore for these instrument modes the direction in inertial space, for a given tangent altitude, is given in the swath template file.

7.3.2.4 Instrument Swath Definition File

Format is described in Appendix A. Each swath calculation mode has its own set of parameters, so the format of the file is calculation mode dependent.

7.3.3 Calling sequence pg_genswath_ext_cycle

For C programs, the call to **pg_genswath_ext_cycle** is (input parameters are underlined):

```
#include ”ppf_genref.h”
{
      long         requested_orbit, flag;
      long         repeat_cycle, cycle_length ;
      long         status, ierr[10];
      double       mlst_drift;
      char         *orbit_scenario_file, *swath_definition_file ;
      char         *instr_swath_file_suffix ;

      status = pg_genswath_ext_cycle (
                        orbit_scenario_file, &requested_orbit, &flag,
                        swath_definition_file, instr_swath_file_suffix,
                        &repeat_cycle,&cycle_length,
                        &mlst_drift, ierr);

/* test status */
}
```

For FORTRAN programs **pg_genswath_ext_cycle** has the following calling sequence (input parameters are underlined, note that the C preprocessor must be used because of the presence of the #include statement):

```
#include ”ppf_genswath.inc”
       INTEGER*4   REQUESTED_ORBIT, FLAG
       INTEGER*4   FIRST_ORBIT, LAST_ORBIT, REPEAT_CYCLE, CYCLE_LENGTH
       INTEGER*4   STATUS, IERR(10)
       REAL*8      MLST_DRIFT
       CHARACTER*(*)    ORBIT_SCENARIO_FILE, SWATH_DEFINITON_FILE
       CHARACTER*(*)    INSTR_SWATH_FILE_SUFFIX

       STATUS = PG_GENSWATH_EXT_CYCLE (
      &                 ORBIT_SCENARIO_FILE, REQUESTED_ORBIT,FLAG
      &                 SWATH_DEFINITON_FILE, INSTR_SWATH_FILE_SUFFIX,
      &                 REPEAT_CYCLE, CYCLE_LENGTH, MLST_DRIFT, IERR)


C test status
```

7.3.4 Input parameters pg_genswath_ext_cycle

c name	c type	Description	Units	Range
orbit_scenario_file	char *	Orbit Scenario File, defining the Envisat-1 reference orbit, by describing the orbits with a changing orbit characteristic.		
requested_orbit	long	Orbit for which the instrument swath template file will be calculated.	absolute orbit number	= 0 or >= start_osf <= stop_osf
flag	long	Flag. If one, 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). The orbit stored in Swath file will be the one given in requested_orbit.		= 0 or 1
swath_definition_file	char *	File name of the instrument swath definition file. Format see section 7.2.2.4.		
instrument_swath_file_suffix	char *	name suffix for output swath file. Prefix if deterministic, based on swath name (6 characters, see Table A.3) and cycle length (4 characters, leading underscores "_" is needed): Example: ASARGS_501 If empty (i.e. ""), the software will generate the suffix according to file name specification presented in RD 4		
Repeat Cycle	long	Repeat Cycle Only used if the flag is set to zero.	days	
Cycle Length	long	Cycle Length of requested_orbit lOnly used if the flag is set to zero.	orbits	
mlst drift	double	Mean Local Solar Time of requested orbit Only used if the flag is set to zero.	sec/year	

Table 5: Input parameters for pg_genswath_ext_cycle

7.3.5 Output parameters pg_genswath_ext_cycle

c name	c type	Array Element	Description	Unit	Range
pg_genswath_ext_cycle	long		Function status flag, 0 = No error > 0 Warnings, results generated < 0 Error, no results generated		
ierr[10]	long		Error status flags		

Table 6: Output parameters for pg_genswath_ext_cycle

7.3.6 Warnings and errors

Next table lists the possible error messages that can be returned by the **pg_genswath_ext_cycle** CFI function after translating the returned status vector into the equivalent list of error messages by calling the function of the PPF_GENREF software library **pg_vector_msg** (see RD 3).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation, 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 **pg_genref** CFI function by calling the function of the PPF_GENREF software library **pg_vector_code** (see RD 3).

Error type	Error message	Cause and impact	Error Code	Error No
ERR	Cannot open Swath Definition File	Computation not performed	PG_CFI_GENSWATH_EXT_CYCLE_OPEN_SDF_ERR	0
ERR	Unable to store Swath Template File names (not enough memory).	Computation not perfomed	PG_CFI_GENSWATH_EXT_CYCLE_STF_FILENAMES_ERR	1
ERR	Error in "SWATH=". There is no cycle length value. Swath Definition File record: %ld.	Computation not perfomed	PG_CFI_GENSWATH_SWATH_EXT_CYCLE_ID_CYCL_LENGTH_1ST_ERR	2
ERR	Cycle length value in "SWATH=" does not match with cycle length for computations. Swath Definition File record: %ld	Computation not perfomed	PG_CFI_GENSWATH_SWATH_EXT_CYCLE_ID_CYCL_LENGTH_2ND_ERR	3
ERR	Cannot open Swath Template File for writting. Swath Template File: %ld failed.	Computation not perfomed	PG_CFI_GENSWATH_EXT_CYCLE_OPEN_STF_WRITE_ERR	4
ERR	Error while writing beginning of Swath Template File. Swath Template File: %ld failed.	Computation not perfomed	PG_CFI_GENSWATH_EXT_CYCLE_STF_BEG_WRITE_ERR	5
ERR	Error while writing comment line. Swath Template File: %ld failed.	Computation not perfomed	PG_CFI_GENSWATH_EXT_CYCLE_COMM_LINE_ERR	6
ERR	Cannot rename Swath Template File: %ld.	Computation not perfomed	PG_CFI_GENSWATH_EXT_CYCLE_RENAME_FILES_ERR	7
ERR	Error in Genswath inputs (REPEAT CYCLE, CYCLE LENGTH or Orbit Scenario File inconsistency).	Computation not perfomed	PG_CFI_GENSWATH_EXT_CYCLE_PG_CHECK_GENSW_INP_ERR	32
ERR	Error in Genswath_ext_cycle input requested orbit. It must be at least one	Computation not perfomed	PG_CFI_GENSWATH_EXT_CYCLE_PG_REQUESTED_ORBIT_ERR	33

Error type	Error message	Cause and impact	Error Code	Error No
ERR	Error while generating Ascending Node parameters.	Computation not perfomed	PG_CFI_GENSWATH_EXT_CYCLE_PG_GENSTATE_ERR	34
ERR	Error in Genswath inputs. [PG]	Computation not perfomed	PG_CFI_GENSWATH_EXT_CYCLE_PG_SDF_PREV_CHECK_ERR	35
ERR	“SWATH=“ value of Swath record: %ld is repeated below.	Computation not perfomed	PG_CFI_GENSWATH_EXT_CYCLE_PG_SDF_PREV_CHECK_SWATH_VALUE_ERR	36
ERR	Error reading Swath record: %ld.	Computation not perfomed	PG_CFI_GENSWATH_EXT_CYCLE_PG_SDF_PREV_CHECK_SWATH_RECORD_ERR	37
ERR	Error reading Swath Definition File header.	Computation not perfomed	PG_CFI_GENSWATH_EXT_CYCLE_PG_SDF_READ_HEADER_ERR	38
ERR	Error reading Swath Definition File record: %ld. [PG]	Computation not perfomed	PG_CFI_GENSWATH_EXT_CYCLE_PG_SDF_REC_READ_ERR	39
ERR	Error in algorithm computations. Swath Template File: %ld failed.	Computation not perfomed	PG_CFI_GENSWATH_EXT_CYCLE_PG_ALGOR_ERR	40
ERR	Error while generating output filename. Swath Template File: %ld failed.	Computation not perfomed	PG_CFI_GENSWATH_EXT_CYCLE_PG_FILENAME_GENER_SWATH_ERR	41
ERR	Error writting fixed header. Swath Template File: %ld failed.	Computation not perfomed	PG_CFI_GENSWATH_EXT_CYCLE_PG_FHR_WRITE_ERR	42
ERR	Error while writing Swath Template File variable header. Swath Template File: %ld failed.	Computation not perfomed	PG_CFI_GENSWATH_EXT_CYCLE_PG_STF_VHR_WRITE_ERR	43

7.3.7 Runtime performances

The following runtime performances have been measured:

Swath Algorithm (Instrument)	Points	Ultra Sparc [sec]
Line (MERIS_)	1200	13.0
Limb (MIP_RH)	1200	18.0
Point (RA_2__)	1200	6.0
ASAR (SA1IM_)	1200	20.0
Inertial (MIPIRH)	1200	12.0

7.3.8 Executable Program genswath_ext_cyc

the **genswath_ext_cyc** executable program can be called from a Unix shell as:

```
genswath_ext_cyc {[-osf    orbit_scenario_file_name
                   -orbit  orbit_number ]  |
                  [-recyc  repeat_cycle
                   -cyclen cycle_length
                   -mlstd  mlst_drift
                   -orbit  orbit_number] }
                   -sdf    swath_definition_file_name
                  [-isfx   instrument_swath_file_name_suffix ]
                  [ -v ]
                  [ -pl_v ]
                  [ -po_v ]
                  [ -pp_v ]
                  [ -pg_v ]
                  [ -help ]
```

Note that:

- order of parameters does not matter
- bracketed parameters are not mandatory (check previous paragraphs for behaviour when not provided)
- Options between curly brackets and separated by a vertical bar are mutually exclusive.
- -pl_v for PPF_LIB verbose mode
- -po_v for PPF_ORBIT verbose mode
- -pp_v for PPF_POINTING verbose mode
- -pg_v for PPF_GENREF verbose mode
- -v for Verbose mode for the four libraries (default is Silent)
- -help option will result in printing the above text on stderr (no execution)
- all errors during execution will be printed in clear messages on stderr

7.4 pg_genosf_ext_cyc

7.4.1 Overview

Before describing the CFI function **pg_genosf_ext_cyc**, it is neccesary to define the following concepts:

We call the “internal” orbit scenario file (internal OSF) to an orbit scenario file that contains orbital changes with the following characteristics:

- The first orbital change begin with the first orbit of the Envisat Mission.
- The file extends for the whole Envisat Mission.
- The orbital changes define the Envisat orbit for the phases before and after the mission extension.
- The orbital changes extend for several cycles.
- The orbital changes for the mission extension phase have MLST drifting parameters (MLST linear, quadratic and harmonic terms)

The **pg_genosf_ext_cyc** function generates an orbit scenario file (official OSF) expanding an input “internal” OSF. This “expansion” consist in the following:

- The orbital changes from the internal OSF previous to the mission extension, are copied to the the official OSF without any change.
- The criteria to know if the OSF record is previous to the mission extension is that the MLST quadratic and harmonic terms are zero (even if the linear term is different from zero, the orbital change is copied to the official OSF without any change).
- The orbital changes from the internal OSF corresponding to the mission extension are populated so that a new orbital change is introduced for every cycle. These new orbital changes will have a constant MLST drift (quadratics and harmonic terms are set to zero). The MLST drift is computed as the average of the MLST drift computed for the cycle.
- The official OSF is generated for a requested orbit interval (from an start_orbit to an stop_orbit)

7.4.2 Calling sequence pg_genosf_ext_cyc

For C programs, the call to **pg_genosf_ext_cyc** is (input parameters are underlined):

```
#include <ppf_genref.h>
{
       long    status, ierr[PG_GENOSF_EXT_CYC_MAX_IERR];
       char     *osf_in, *osf_out;
       long     *start_orb, long *stop_orb,

       status = pg_genosf_ext_cyc(osf_in, &start_orb, &stop_orb,
                                  osf_out, ierr)
       /* test status */
}
```

7.4.3 Input parameters pg_genosf_ext_cyc

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
osf_in	char *	-	Input Orbit Scenario File, defining the Envisat-1 reference orbit, by describing the orbits with a changing orbit characteristic.		
start_orb	long	-	First start orbit for the Orbit Scenario file. If set to zero, the first orbit of the osf_in is taken.	-	>=0
stop_orb	long	-	First orbit of the last cycle to be included in the Orbit Scenario file		
osf_out	char*	-	Output OSF filename	-	-

Table 7: Input parameters for pg_genosf_ext_cyc

7.4.4 Output parameters pg_genosf_ext_cyc

c name	c type	Array Element	Description	Unit	Range
pg_genosf_ext_cyc	long		Function status flag, 0 = No error > 0 Warnings, results generated < 0 Error, no results generated		
ierr[]	long		Error status flags		

Table 8: Output parameters for pg_genosf_ext_cyc

7.4.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **pg_genosf_ext_cyc** CFI function after translating the returned status vector into the equivalent list of error messages by calling the function of the PPF_GENREF software library **pg_vector_msg** (see RD 3).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation, 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 **pg_genosf_ext_cyc** CFI function by calling the function of the PPF_GENREF software library **pg_vector_code** (see RD 3).

Error type	Error message	Cause and impact	Error Code	Error No
ERR	Could not open input file	Computation not performed	PG_CFI_GENOSF_EXT_CYC_OSF_OPEN_ERR	0
ERR	Could not read header from input OSF file	Computation not perfomed	PG_CFI_GENOSF_EXT_CYC_OSF_READ_HDR_ERR	1
ERR	Could not read Data Block from input OSF file	Computation not perfomed	PG_CFI_GENOSF_EXT_CYC_OSF_READ_DBL_ERR	2
ERR	Could not write output OSF	Computation not perfomed	PG_CFI_GENOSF_EXT_CYC_OSF_WRITE_ERR	3
ERR	Memory allocation error	Computation not perfomed	PG_CFI_GENOSF_EXT_CYC_MEMORY_ERR	4
ERR	Wrong input start orbit	Computation not perfomed	PG_CFI_GENOSF_EXT_CYC_WRONG_START_ORBIT_ERR	5
ERR	Error in PO_Genstate	Computation not perfomed	PG_CFI_GENOSF_EXT_CYC_GENSTATE_ERR	6

7.4.6 Runtime performances

The following runtime performances have been measured:

Ultra Sparc [sec]
TBD

8 LIBRARY PRECAUTIONS

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

- When a message like

  PPF_GENREF >>> ERROR in *pg_function*: Internal computation error # *n*

  or

  PPF_GENREF >>> WARNING in *pg_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.

- In some cases, the string [PL] , [PO], [PP] or [PG] appears at the end of the error/warning message. In these cases, run the program in *pl_verbose*, *po_verbose*, *pp_verbose* or *pg_verbose* mode for a complete description of warnings and errors coming from other libraries.

- The number of characters between quotes in input files is limited to 256. The maximum number of characters for a line in an input file (before new line character) is 512.

9 KNOWN PROBLEMS

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

CFI library	Problem	Work around solution
	(no known problems)	

APPENDIX. SWATH DEFINITION FILE

A. Swath Definition File

This file is only used by the PPF_GENREF CFI software in order to produce the Swath Template Files

File type comment (in 1st line of file):

```
; Swath Definition File
```

A. 1 FHR

Follows the format described in section *4.3.2* in RD 4 with:

- File ID (in file name) = `MPL_SW_DEF`
- originator (in file name) = `EMM`
- DESTINATION = `TBD`

A.2 VHR

Table A.1 detail the Swath Template File VHR format.

Table A.1 Swath Definition File VHR Format

N	Description	units	Byte Length	Data Type	C Format
1	**RECORD**	keyword	6	string	%6s
	blank space		1	string	%1s
	sdf_vhr	keyword	7	string	%7s
	blank space		1	string	%1s
	; Variable Header	comment		string	%s
	newline character	terminator	1	string	\n
2	newline character	empty line	1	string	\n
3	**REPEAT_CYCLE=**	keyword	13	string	%13s
	Repeat Cycle		4	+xxx	%+04d
	<days>	unit	6	string	%6s
	newline character	terminator	1	string	\n
4	**CYCLE_LENGTH=**	keyword	13	string	%13s
	Cycle Length		6	+xxxxx	%+06d
	<orbits>	unit	8	string	%8s
	newline character	terminator	1	string	\n

Table A.1 Swath Definition File VHR Format

N	Description	units	Byte Length	Data Type	C Format
5	newline character	empty line	1	string	\n
6	**RECORD**	keyword	6	string	%6s
	blank space		1	string	%1s
	aocs_ampl:	keyword	10	string	%10s
	blank space	separator	1	string	%1s
	PITCH=	keyword	6	string	%6s
	Pitch Amplitude		11	+xxx.xxxxxx	%#+011.6f
	<deg>	unit	5	string	%5s
	blank space	separator	1	string	%1s
	ROLL=	keyword	5	string	%5s
	Roll Amplitude		11	+xxx.xxxxxx	%#+011.6f
	<deg>	unit	5	string	%5s
	blank space	separator	1	string	%1s
	YAW=	keyword	4	string	%4s
	Yaw Amplitude	metres	11	+xxxxxx.xxx	%#+011.3f
	<deg>	unit	5	string	%5s
	blank space	separator	1	string	\n
	ENDRECORD	keyword	9	string	%9s
	newline character	terminator	1	string	\n
7	**NUM_SWATH=**	keyword	10	string	%10s
	Number of Swath Files Defined		4	+xxx	%+04d
	newline character	terminator	1	string	\n
8	**SDF_VERSION=**	keyword	12	string	%12s
	quotation mark	-	1	string	\”
	Swath definition file version		2	xx	%02ld
	quotation mark	-	1	string	\”
	newline character	terminator	1	string	\n
9	**STF_FREQ=**	keyword	9	string	%9s
	Swath template file frequency		4	+xxx	%+04d
	newline character	terminator	1	string	\n
10	newline character	empty line	1	string	\n

Table A.1 Swath Definition File VHR Format

N	Description	units	Byte Length	Data Type	C Format
11	**ENDRECORD**	keyword	9	string	%9s
	blank space		1	string	%1s
	sdf_vhr	keyword	7	string	%7s
	newline character	terminator	1	string	\n

A.3 Data Block

The file will consist of 1 list of variable length:

- Swath Files Definitions.

The structure contains among other details 3 sub-structures which all have variable format:

- geometry: azimuth/elevation/altitude values. 3 cases:
  - line: values for left, mid and right points
  - point: values for 1 point only
  - limb: same as line, but no elevation values (tangent point is used)
- mispointing: pitch/roll/yaw values. 3 cases:
  - all: values for bias and harmonics
  - bias: values for bias only
  - none: no values (i.e. no mispointing)
- specific to ASAR: window/pulse values. 3 cases:
  - wide: values for left and right points
  - narrow: values for 1 point only
  - none: no values (i.e. swath is not an ASAR swath)

Table A.2, Table A.3, Table A.4, Table A.5, Table A.6, Table A.7, Table A.8, Table A.9, Table A.10, Table A.11 and Table A.12 detail the Swath Definition File data block format.

Table A.2 Swath Definition File data block Format

N	Description	units	Byte Length	Data Type	C Format
1	**LIST**	keyword	4	string	%4s
	blank space		1	string	%1s
	num_swath=	keyword	14	string	%14s
	Number of Swath Template Files required		5	xxxxx	%05d
	blank space		1	string	%1s
	; Swaths Required	comment		string	%s
	newline character	terminator	1	string	\n
2	newline character	empty line	1	string	\n
3	list of **num_swath** Swath Definitions (see format below) all separated by empty lines				
4	newline character	empty line	1	string	\n
5	**ENDLIST**	keyword	7	string	%7s
	blank space		1	string	%1s
	num_swath	keyword	13	string	%13s
	newline character	terminator	1	string	\n

Table A.3 Swath Definition Format

N	Description	units	Byte Length	Data Type	C Format
1	**RECORD**	keyword	6	string	%6s
	blank space		1	string	%1s
	swath	keyword	5	string	%5s
	newline character	terminator	1	string	\n

Table A.3 Swath Definition Format

N	Description	units	Byte Length	Data Type	C Format
2	2 blank spaces	indentation	2	string	%2s
	SWATH_DESCRIPTOR=	keyword	17	string	%17s
	quotation mark	-	1	string	\”
	Swath Descriptor (for information) ASCII string describing the swath (unused characters are blanked)	-	28	string	%28s
	quotation mark	-	1	string	\”
	newline character	terminator	1	string	\n
3	2 blank spaces	indentation	2	string	%2s
	SWATH=	keyword	5	string	%5s
	quotation mark		1	string	\”
	Swath ID (used by software, must be unique) see section 7.2.2 Note that the table gives the list of “standard” Envisat swaths for information. It does not restrict the values for this parameter.		6	string	%6s
	quotation mark		1	string	\”
	newline character	terminator	1	string	\n
4	2 blank spaces	indentation	2	string	%2s
	ALGORITHM=	keyword	10	string	%10s
	quotation mark		1	string	\”
	Algorithm to use one of: - **line** - **point** - **limb** - **inertial** - **asar**		8	string	%8s
	quotation mark		1	string	\”
	newline character	terminator	1	string	\n

Table A.3 Swath Definition Format

N	Description	units	Byte Length	Data Type	C Format
5	2 blank spaces	indentation	2	string	%2s
	RECORD	keyword	6	string	%6s
	blank space		1	string	%1s
	refraction:	keyword	11	string	%11s
	blank space	separator	1	string	%1s
	MODEL=	keyword	6	string	%6s
	quotation mark	-	1	string	\”
	Refraction Model used one of (right-filled with blanks if needed): - **NO_REF** - **STD_REF** - **USER_REF** - **PRED_REF**		8	string	%8s
	quotation mark	-	1	string	\”
	blank space	separator	1	string	%1s
	FREQ=	keyword	5	string	%5s
	Frequency	MHz	11	+xxxxxxxxxx	%+011d
	<Mhz>	unit	5	string	%5s
	blank space	separator	1	string	%1s
	ENDRECORD	keyword	9	string	%9s
	newline character	terminator	1	string	\n
6	2 blank spaces	indentation	2	string	%2s
	NUM_SWATH_REC=	keyword	14	string	%14s
	Number of swath records required		6	+xxxxx	%+06d
	newline character	terminator	1	string	\n
7	2 blank spaces	indentation	2	string	%2s
	UNION	keyword	5	string	%5s
	blank space		1	string	%1s
	geometry=	keyword	9	string	%9s
	Geometry Record structure name one of: - **line_geometry** **- point_geometry** - **limb_geometry**		18	string	%18s
	newline character	terminator	1	string	\n

Table A.3 Swath Definition Format

N	Description	units	Byte Length	Data Type	C Format
8	one of **line_geometry**, **point_geometry** of **limb_geometry** records (see description below)				
9	2 blank spaces	indentation	2	string	%2s
	ENDUNION	keyword	8	string	%8s
	blank space		1	string	%1s
	geometry	keyword	8	string	%8s
	newline character	terminator	1	string	\n
10	2 blank spaces	indentation	2	string	%2s
	UNION	keyword	5	string	%5s
	blank space		1	string	%1s
	mispointing=	keyword	12	string	%12s
	Mispointing Record structure name one of: - **all_mispointing** **- bias_mispointing** - **no_mispointing**		18	string	%18s
	newline character	terminator	1	string	\n
11	one of **all_mispointing**, **bias_mispointing** of **no_mispointing** records (see description below)				
12	2 blank spaces	indentation	2	string	%2s
	ENDUNION	keyword	8	string	%8s
	blank space		1	string	%1s
	mispointing	keyword	11	string	%11s
	newline character	terminator	1	string	\n
13	2 blank spaces	indentation	2	string	%2s
	UNION	keyword	5	string	%5s
	blank space		1	string	%1s
	spec_asar=	keyword	10	string	%10s
	Specific ASAR Record structure name one of: - **wide_asar** **- narrow_asar** - **no_asar**		18	string	%18s
	newline character	terminator	1	string	\n
14	one of **wide_asar**, **narrow_asar** of **no_asar** records (see description below)				

Table A.3 Swath Definition Format

N	Description	units	Byte Length	Data Type	C Format
15	2 blank spaces	indentation	2	string	%2s
	ENDUNION	keyword	8	string	%8s
	blank space		1	string	%1s
	spec_asar	keyword	9	string	%9s
	newline character	terminator	1	string	\n
16	**ENDRECORD**	keyword	9	string	%9s
	blank space		1	string	%1s
	swath	keyword	5	string	%5s
	newline character	terminator	1	string	\n

Table A.4 **line_geometry** record format

N	Description	units	Byte Length	Data Type	C Format
1	4 blank spaces	indentation	2	string	%2s
	RECORD	keyword	6	string	%6s
	2 blank spaces		2	string	%2s
	left_pt:	keyword	8	string	%8s
	blank space	separator	1	string	%1s
	AZ=	keyword	3	string	%3s
	Azimuth of left point		11	+xxx.xxxxxx	%#+011.6f
	<deg>	unit	5	string	%5s
	blank space	separator	1	string	%1s
	EL=	keyword	3	string	%3s
	Elevation of left point		11	+xxx.xxxxxx	%#+011.6f
	<deg>	unit	5	string	%5s
	blank space	separator	1	string	%1s
	ALT=	keyword	4	string	%4s
	Altitude of left point	metres	11	+xxxxxx.xxx	%#+011.3f
	<m>	unit	3	string	%3s
	blank space	separator	1	string	\n

Table A.4 **line_geometry** record format

N	Description	units	Byte Length	Data Type	C Format
	ENDRECORD	keyword	9	string	%9s
	newline character	terminator	1	string	\n
2	4 blank spaces	indentation	2	string	%2s
	RECORD	keyword	6	string	%6s
	3 blank spaces		3	string	%3s
	mid_pt:	keyword	7	string	%7s
	blank space	separator	1	string	%1s
	AZ=	keyword	3	string	%3s
	Azimuth of mid point		11	+xxx.xxxxxx	%#+011.6f
	<deg>	unit	5	string	%5s
	blank space	separator	1	string	%1s
	EL=	keyword	3	string	%3s
	Elevation of mid point		11	+xxx.xxxxxx	%#+011.6f
	<deg>	unit	5	string	%5s
	blank space	separator	1	string	%1s
	ALT=	keyword	4	string	%4s
	Altitude of mid point	metres	11	+xxxxxx.xxx	%#+011.3f
	<m>	unit	3	string	%3s
	blank space	separator	1	string	\n
	ENDRECORD	keyword	9	string	%9s
	newline character	terminator	1	string	\n
3	4 blank spaces	indentation	2	string	%2s
	RECORD	keyword	6	string	%6s
	blank space		1	string	%1s
	right_pt:	keyword	9	string	%9s
	blank space	separator	1	string	%1s
	AZ=	keyword	3	string	%3s
	Azimuth of right point		11	+xxx.xxxxxx	%#+011.6f
	<deg>	unit	5	string	%5s
	blank space	separator	1	string	%1s

Table A.4 **line_geometry** record format

N	Description	units	Byte Length	Data Type	C Format
	EL=	keyword	3	string	%3s
	Elevation of right point		11	+xxx.xxxxxx	%#+011.6f
	<deg>	unit	5	string	%5s
	blank space	separator	1	string	%1s
	ALT=	keyword	4	string	%4s
	Altitude of right point	metres	11	+xxxxxx.xxx	%#+011.3f
	<m>	unit	3	string	%3s
	blank space	separator	1	string	\n
	ENDRECORD	keyword	9	string	%9s
	newline character	terminator	1	string	\n

Table A.5 **point_geometry** record format

N	Description	units	Byte Length	Data Type	C Format
1	4 blank spaces	indentation	2	string	%2s
	RECORD	keyword	6	string	%6s
	blank space		1	string	%1s
	pt:	keyword	3	string	%3s
	blank space	separator	1	string	%1s
	AZ=	keyword	3	string	%3s
	Azimuth of point		11	+xxx.xxxxxx	%#+011.6f
	<deg>	unit	5	string	%5s
	blank space	separator	1	string	%1s
	EL=	keyword	3	string	%3s
	Elevation of point		11	+xxx.xxxxxx	%#+011.6f
	<deg>	unit	5	string	%5s
	blank space	separator	1	string	%1s
	ALT=	keyword	4	string	%4s
	Altitude of point	metres	11	+xxxxxx.xxx	%#+011.3f
	<m>	unit	3	string	%3s
	blank space	separator	1	string	\n
	ENDRECORD	keyword	9	string	%9s
	newline character	terminator	1	string	\n

Table A.6 **limb_geometry** record format

N	Description	units	Byte Length	Data Type	C Format
1	4 blank spaces	indentation	2	string	%2s
	RECORD	keyword	6	string	%6s
	2 blank spaces		2	string	%2s
	left_pt:	keyword	8	string	%8s
	blank space	separator	1	string	%1s
	AZ=	keyword	3	string	%3s
	Azimuth of left point		11	+xxx.xxxxxx	%#+011.6f
	<deg>	unit	5	string	%5s
	blank space	separator	1	string	%1s
	ALT=	keyword	4	string	%4s
	Altitude of left point	metres	11	+xxxxxx.xxx	%#+011.3f
	<m>	unit	3	string	%3s
	blank space	separator	1	string	\n
	ENDRECORD	keyword	9	string	%9s
	newline character	terminator	1	string	\n
2	4 blank spaces	indentation	2	string	%2s
	RECORD	keyword	6	string	%6s
	3 blank spaces		3	string	%3s
	mid_pt:	keyword	7	string	%7s
	blank space	separator	1	string	%1s
	AZ=	keyword	3	string	%3s
	Azimuth of mid point		11	+xxx.xxxxxx	%#+011.6f
	<deg>	unit	5	string	%5s
	blank space	separator	1	string	%1s
	ALT=	keyword	4	string	%4s
	Altitude of mid point	metres	11	+xxxxxx.xxx	%#+011.3f
	<m>	unit	3	string	%3s
	blank space	separator	1	string	\n

Table A.6 **limb_geometry** record format

N	Description	units	Byte Length	Data Type	C Format
	ENDRECORD	keyword	9	string	%9s
	newline character	terminator	1	string	\n
3	4 blank spaces	indentation	2	string	%2s
	RECORD	keyword	6	string	%6s
	blank space		1	string	%1s
	right_pt:	keyword	9	string	%9s
	blank space	separator	1	string	%1s
	AZ=	keyword	3	string	%3s
	Azimuth of right point		11	+xxx.xxxxxx	%#+011.6f
	<deg>	unit	5	string	%5s
	blank space	separator	1	string	%1s
	ALT=	keyword	4	string	%4s
	Altitude of right point	metres	11	+xxxxxx.xxx	%#+011.3f
	<m>	unit	3	string	%3s
	blank space	separator	1	string	\n
	ENDRECORD	keyword	9	string	%9s
	newline character	terminator	1	string	\n

Table A.7 **all_mispointing** record format

N	Description	units	Byte Length	Data Type	C Format
1	4 blank spaces	indentation	2	string	%2s
	RECORD	keyword	6	string	%6s
	blank space		1	string	%1s
	bias:	keyword	5	string	%5s
	blank space	separator	1	string	%1s
	PITCH=	keyword	6	string	%6s
	Pitch Bias		11	+xxx.xxxxxx	%#+011.6f
	<deg>	unit	5	string	%5s
	blank space	separator	1	string	%1s
	ROLL=	keyword	5	string	%5s
	Roll Bias		11	+xxx.xxxxxx	%#+011.6f
	<deg>	unit	5	string	%5s
	blank space	separator	1	string	%1s
	YAW=	keyword	4	string	%4s
	Yaw Bias	metres	11	+xxxxxx.xxx	%#+011.3f
	<deg>	unit	5	string	%5s
	blank space	separator	1	string	\n
	ENDRECORD	keyword	9	string	%9s
	newline character	terminator	1	string	\n
2	4 blank spaces	indentation	2	string	%2s
	RECORD	keyword	6	string	%6s
	2 blank spaces		2	string	%2s
	sin:	keyword	4	string	%4s
	blank space	separator	1	string	%1s
	PITCH=	keyword	6	string	%6s
	Pitch Bias		11	+xxx.xxxxxx	%#+011.6f
	<deg>	unit	5	string	%5s
	blank space	separator	1	string	%1s

Table A.7 **all_mispointing** record format

N	Description	units	Byte Length	Data Type	C Format
	ROLL=	keyword	5	string	%5s
	Roll Bias		11	+xxx.xxxxxx	%#+011.6f
	<deg>	unit	5	string	%5s
	blank space	separator	1	string	%1s
	YAW=	keyword	4	string	%4s
	Yaw Bias	metres	11	+xxxxxx.xxx	%#+011.3f
	<deg>	unit	5	string	%5s
	blank space	separator	1	string	\n
	ENDRECORD	keyword	9	string	%9s
	newline character	terminator	1	string	\n
3	4 blank spaces	indentation	2	string	%2s
	RECORD	keyword	6	string	%6s
	2 blank spaces		2	string	%2s
	cos:	keyword	4	string	%4s
	blank space	separator	1	string	%1s
	PITCH=	keyword	6	string	%6s
	Pitch Bias		11	+xxx.xxxxxx	%#+011.6f
	<deg>	unit	5	string	%5s
	blank space	separator	1	string	%1s
	ROLL=	keyword	5	string	%5s
	Roll Bias		11	+xxx.xxxxxx	%#+011.6f
	<deg>	unit	5	string	%5s
	blank space	separator	1	string	%1s
	YAW=	keyword	4	string	%4s
	Yaw Bias	metres	11	+xxxxxx.xxx	%#+011.3f
	<deg>	unit	5	string	%5s
	blank space	separator	1	string	\n
	ENDRECORD	keyword	9	string	%9s
	newline character	terminator	1	string	\n

Table A.8 **bias_mispointing** record format

N	Description	units	Byte Length	Data Type	C Format
1	4 blank spaces	indentation	2	string	%2s
	RECORD	keyword	6	string	%6s
	blank space		1	string	%1s
	bias:	keyword	5	string	%5s
	blank space	separator	1	string	%1s
	PITCH=	keyword	6	string	%6s
	Pitch Bias		11	+xxx.xxxxxx	%#+011.6f
	<deg>	unit	5	string	%5s
	blank space	separator	1	string	%1s
	ROLL=	keyword	5	string	%5s
	Roll Bias		11	+xxx.xxxxxx	%#+011.6f
	<deg>	unit	5	string	%5s
	blank space	separator	1	string	%1s
	YAW=	keyword	4	string	%4s
	Yaw Bias	metres	11	+xxxxxx.xxx	%#+011.3f
	<deg>	unit	5	string	%5s
	blank space	separator	1	string	\n
	ENDRECORD	keyword	9	string	%9s
	newline character	terminator	1	string	\n

Table A.9 **no_mispointing** record format

N	Description	units	Byte Length	Data Type	C Format
	this record is empty by definition				

Table A.10 **wide_asar** record format

N	Description	units	Byte Length	Data Type	C Format
1	4 blank spaces	indentation	2	string	%2s
	RECORD	keyword	6	string	%6s
	2 blank spaces		2	string	%2s
	left_asar:	keyword	10	string	%10s
	blank space	separator	1	string	%1s
	WINDOW=	keyword	7	string	%7s
	Window of left point		11	+xxx.xxx	%#+08.3f
	<10-6s>	unit	7	string	%7s
	blank space	separator	1	string	%1s
	PULSE=	keyword	6	string	%6s
	Pulse of left point		11	+xxx.xxx	%#+08.3f
	<10-6s>	unit	7	string	%7s
	blank space	separator	1	string	%1s
	ENDRECORD	keyword	9	string	%9s
	newline character	terminator	1	string	\n
2	4 blank spaces	indentation	2	string	%2s
	RECORD	keyword	6	string	%6s
	blank space		1	string	%1s
	right_asar:	keyword	11	string	%11s
	blank space	separator	1	string	%1s
	WINDOW=	keyword	7	string	%7s
	Window of right point		11	+xxx.xxx	%#+08.3f
	<10-6s>	unit	7	string	%7s
	blank space	separator	1	string	%1s
	PULSE=	keyword	6	string	%6s
	Pulse of right point		11	+xxx.xxx	%#+08.3f
	<10-6s>	unit	7	string	%7s
	blank space	separator	1	string	%1s

Table A.10 **wide_asar** record format

N	Description	units	Byte Length	Data Type	C Format
	ENDRECORD	keyword	9	string	%9s
	newline character	terminator	1	string	\n

Table A.11 **narrow_asar** record format

N	Description	units	Byte Length	Data Type	C Format
1	4 blank spaces	indentation	2	string	%2s
	RECORD	keyword	6	string	%6s
	blank space		1	string	%1s
	asar:	keyword	5	string	%5s
	blank space	separator	1	string	%1s
	WINDOW=	keyword	7	string	%7s
	Window of right point		11	+xxx.xxx	%#+08.3f
	<10-6s>	unit	7	string	%7s
	blank space	separator	1	string	%1s
	PULSE=	keyword	6	string	%6s
	Pulse of right point		11	+xxx.xxx	%#+08.3f
	<10-6s>	unit	7	string	%7s
	blank space	separator	1	string	%1s
	ENDRECORD	keyword	9	string	%9s
	newline character	terminator	1	string	\n

Table A.12 **no_asar** record format

N	Description	units	Byte Length	Data Type	C Format
	this record is empty by definition				

A.4 Example

An example Swath Definition File is shown in next figure.

Figure 2 Example Swath Definition File

```
FILE ; Swath Definition File
;--------------------------------------------------
RECORD fhr ; Fixed Header

FILENAME=”MPL_SW_DEFTEMM19970327_160000_00000000_00000001_yyyymmdd_hhmmss_yyyymmdd_hhmmss.N1”

DESTINATION=”.........”
PHASE_START=+xxx
CYCLE_START=+xxx
REL_START_ORBIT=+xxxxx
ABS_START_ORBIT=+xxxxx

ENDRECORD fhr
;--------------------------------------------------
RECORD sdf_vhr ; Variable Header

REPEAT_CYCLE=+xxx<days>
CYCLE_LENGTH=+xxxxx<orbits>

NUM_SWATH=+xxx

SDF_VERSION="02"
STF_FREQ=+004

ENDRECORD sdf_vhr
;--------------------------------------------------
LIST num_swath=xxx ; Swaths Required

RECORD swath
  SWATH_DESCRIPTOR=”...........................”
  SWATH=”..........”
  ALGORITHM=”........”
  RECORD refraction: MODEL=”........” FREQ=+xxxxxxxxxx<Mhz> ENDRECORD
  NUM_SWATH_REC=+01200
  UNION geometry=line_geometry
    RECORD  left_pt: AZ=+xxx.xxxxxx<deg> EL=+xxx.xxxxxx<deg> ALT=xxxxxx.xxx<m> ENDRECORD
    RECORD   mid_pt: AZ=+xxx.xxxxxx<deg> EL=+xxx.xxxxxx<deg> ALT=xxxxxx.xxx<m> ENDRECORD
    RECORD right_pt: AZ=+xxx.xxxxxx<deg> EL=+xxx.xxxxxx<deg> ALT=xxxxxx.xxx<m> ENDRECORD
  ENDUNION geometry
  UNION mispointing=all_mispointing
    RECORD bias: PITCH=+xxx.xxxxxx<deg> ROLL=+xxx.xxxxxx<deg> YAW=xxx.xxxxxx<deg> ENDRECORD
    RECORD  sin: PITCH=+xxx.xxxxxx<deg> ROLL=+xxx.xxxxxx<deg> YAW=xxx.xxxxxx<deg> ENDRECORD
    RECORD  cos: PITCH=+xxx.xxxxxx<deg> ROLL=+xxx.xxxxxx<deg> YAW=xxx.xxxxxx<deg> ENDRECORD
  ENDUNION mispointing
  UNION spec_asar=wide_asar
    RECORD  left_asar: WINDOW=+xxx.xxx<10-6s> PULSE=+xxx.xxx<10-6s> ENDRECORD
    RECORD right_asar: WINDOW=+xxx.xxx<10-6s> PULSE=+xxx.xxx<10-6s> ENDRECORD
  ENDUNION asar
ENDRECORD swath

RECORD swath
  SWATH_DESCRIPTOR=”...........................”
  SWATH=”..........”
  ALGORITHM=”........”
  RECORD refraction: MODEL=”........” FREQ=+xxxxxxxxxx<Mhz> ENDRECORD
  NUM_SWATH_REC=+01200
  UNION geometry=inertial_geometry
    RECORD  left_pt: AZ=+xxx.xxxxxx<deg> ALT=xxxxxx.xxx<m> ENDRECORD
    RECORD   mid_pt: AZ=+xxx.xxxxxx<deg> ALT=xxxxxx.xxx<m> ENDRECORD
    RECORD right_pt: AZ=+xxx.xxxxxx<deg> ALT=xxxxxx.xxx<m> ENDRECORD
  ENDUNION geometry
  UNION mispointing=no_mispointing
  ENDUNION mispointing
  UNION spec_asar=no_asar
  ENDUNION asar
ENDRECORD swath

ENDLIST num_swath
```