EARTH OBSERVATION MISSION CFI SOFTWARE

Release Notes –Version 4.7

1 INTRODUCTION

This document describes the changes introduced in this release of the Earth Observation Mission CFI Software.

2 RELEASE DESCRIPTION

2.1 Software

The following table lists the released libraries, their version and issue date:

Library Name	Version	Issue Date
File Handling	4.7	28 March 2014
Data Handling	4.7	28 March 2014
Lib	4.7	28 March 2014
Orbit	4.7	28 March 2014
Pointing	4.7	28 March 2014
Visibility	4.7	28 March 2014
EECommon (*)	4.7	28 March 2014

(*) only C++ and JAVA APIs

The core API of the above libraries is written in C and provides an API for C, C++ and JAVA.

The libraries installation packages are available for download at the following URL (registration required):

http://eop-cfi.esa.int/index.php/mission-cfi-software/eocfi-software/branch-4-x/eocfi-v4x-download

2.2 Documentation

The following documents are available:

Type	Document Name	Version
General	Mission Conventions Document	4.7
General	General Software User Manual	4.7
C API	Quick Start Guide	4.7
C API	File Handling Software User Manual	4.7
C API	Data Handling Software User Manual	4.7
C API	Lib Software User Manual	4.7
C API	Orbit Software User Manual	4.7
C API	Pointing Software User Manual	4.7
C API	Visibility Software User Manual	4.7

The documentation is available for download and on-line browsing at the following URL:
http://eop-cfi.esa.int/index.php/mission-cfi-software/eocfi-software/branch-4-x/eocfi-v4x-documentation

More information on the Earth Observation CFI Software can be found at the following URL:
http://eop-cfi.esa.int/index.php/mission-cfi-software/eocfi-software

2.3 Supported platforms

The following platforms are supported by this release of the CFI
(the following are requirements for the **C API**):

- **LINUX32_LEGACY**
  - LINUX 32-bits (Legacy)
  - Platform Requirements: x86 based PC, Linux Operating System (Kernel version 2.6.x)
  - Software Requirements: gcc compiler version 4.2.x, glibc (C Library) version 2.7

- **LINUX64_LEGACY**
  - LINUX 64-bits (Legacy)
  - Platform Requirements: x86_64 based PC, Linux Operating System (Kernel version 2.6.x)

  - Software Requirements: gcc compiler version 4.2.x, glibc (C Library) version 2.7

- **LINUX64**
  - LINUX 64-bits
  - Platform Requirements: x86_64 based PC, Linux Operating System (Kernel version 2.6.x)
  - Software Requirements: gcc compiler version 4.5.x, glibc (C Library) version 2.12

- **WINDOWS**
  - Microsoft WINDOWS PC (32-bits)
  - Platform Requirements: x86 based PC, Microsoft Windows XP Operating Systems.
  - Software Requirements: Microsoft Visual C++ Compiler (Visual Studio 2008 Professional)

- **MACIN64**
  - MACOSX on Intel (64-bits)
  - Platform Requirements: x86_64 based Mac Computer, Mac OS X version 10.5.x
  - Software Requirements: gcc compiler version 4.2.x

The following are additional requirements for the **C++ API** (a C++ compiler is required):

- g++ compiler version 4.2.x for LINUX32_LEGACY, LINUX64_LEGACY, MACIN64
- g++ compiler version 4.5.x for LINUX64
- Microsoft Visual C++ Compiler (Visual Studio 2008) for WINDOWS

The following are additional requirements for the **JAVA API** (a JAVA SDK is required):

- Java Standard Edition (SE) version 6 for all platforms

2.4 Installation Packages

The CFI libraries are provided as zip packages:

API	Package Name	MD5 Checksum
C	EOCFI-4.7-CLIB-LINUX32_LEGACY.zip	eda3b45ca6b5c0571e5bffddc7609a92
C	EOCFI-4.7-CLIB-LINUX64.zip	650d7e63a509db497905f390310d94f3
C	EOCFI-4.7-CLIB-LINUX64_LEGACY.zip	468c3732d2fc0c2e8b70146155caccdd

C	EOCFI-4.7-CLIB-MACIN64.zip	d5bacdf1b7278dbfb1e8fe0fb792d184
C	EOCFI-4.7-CLIB-WINDOWS.zip	d3a7240ee70f09a20f01e827247e52c5
C++	EOCFI-4.7-CPPLIB-LINUX32_LEGACY.zip	10e2e8d3df134031671961bd597cc204
C++	EOCFI-4.7-CPPLIB-LINUX64.zip	c98d143daf1417572d1fd9373fea3383
C++	EOCFI-4.7-CPPLIB-LINUX64_LEGACY.zip	cfe044f712387980785cbf56b7caccb7
C++	EOCFI-4.7-CPPLIB-MACIN64.zip	7dfb0c9e147d279dfce9308a99e6bf7d
C++	EOCFI-4.7-CPPLIB-WINDOWS_DLL.zip	3a8cbc0ec5e112689f2c07b5be216858
C++	EOCFI-4.7-CPPLIB-WINDOWS_STA.zip	780a3400cc45528ab79b920e02d86306
JAVA	EOCFI-4.7-JAVALIB-LINUX32_LEGACY.zip	41d879d1dad9e8b6453a9110db28d451
JAVA	EOCFI-4.7-JAVALIB-LINUX64.zip	cb03b8c9348957bac42cdb0e322d4f77
JAVA	EOCFI-4.7-JAVALIB-LINUX64_LEGACY.zip	73bb394974a7ba4a87ace1a2aa1b577e
JAVA	EOCFI-4.7-JAVALIB-MACIN64.zip	62eb8bdcfaedfa52ca4c3796dec693a1
JAVA	EOCFI-4.7-JAVALIB-WINDOWS.zip	598e87fac4f01b674afd72f1b0b322a3

(*) Dynamic libraries (DLLs)

(**) Static libraries

DEM datasets are distributed separately and are available for download at the following URL:

http://eop-cfi.esa.int/index.php/mission-cfi-software/eocfi-software/support-files

2.5 Installation Hints

The CFI libraries can be installed by expanding the installation package in any directory.

For specific hints related to the usage of the libraries, please consult Section 6 “CFI LIBRARIES INSTALLATION” of the General SUM and Section 6 “LIBRARY USAGE” of each Library User Manual.

In order to be able to use the XML validation function in the explorer_data_handling library, it is necessary to install the xerces libraries and the SAX2Count binary. The PATH environment variable shall be pointing at the SAX2Count location.

As of version 4.3, dynamic linking to libxml2 external libraries is no longer required.

As of version 4.5, user applications using the pointing library need to be built with openmp support (adding –fopenmp switch in gcc).

NOTE for MACIN64 platform, Xcode users:

Xcode is the suite of Software development tools provided by Apple.

- Xcode 4:
  In Xcode 4. llvm-gcc is the default compiler.
  llvm-gcc can build an application linking against the EOCFI C libraries.

Incompatibilities with C++ and Java APIs have been found due to missing or redefined symbols (e.g. libgomp, the openmp shared library, is not provided).

- Xcode 5:
As of version 5, llvm-gcc has been removed from Xcode and the default compiler is clang. clang can build an application linking against the EOCFI C libraries. However openmp is not supported by clang. Therefore, the –fopenmp shall not be used. Functions using parallelized computations, e.g. xp_target_list... functions cannot be used.
Incompatibilities with C++ and Java APIs have been found due to missing or redefined symbols (e.g. libgomp, the openmp shared library, is not provided).

3 NEW FEATURES

The following sections describe the new features introduced in this release.

The description refers to the C API. Equivalent features and methods are available in the C++ and JAVA APIs. For further details on the presented features, the user may want to refer to the User Manual of the related library.

3.1 General

- **New Missions supported**: **Jason-CS, Metop-SG, Sentinel-5P.**
Mission identifiers for the new supported missions are available.
(this feature has been requested by Jason-CS, Metop-SG and Sentinel-5P projects respectively).

3.2 Data Handling Library

- **New function to read SP3 files: xd_read_sp3**
The xd_read_sp3 CFI function reads a Standard Product 3 C (SP3-C) File and extracts data from the header and Orbit State Vectors. The specification of SP3-C files can be found at the following URL: http://igscb.jpl.nasa.gov/igscb/data/format/sp3c.txt
The new function **xd_free_sp3** can be used to free the memory allocated by xd_read_sp3.
Equivalent methods are available in the C++ and Java API: see **SP3File** Class.
(this feature has been requested by the Metop-SG project).

- **Extended function: xd_read_doris**
This function has been extended to read Doris files for Sentinel-3. The Doris file is composed by a sequence of Doris packets as described in CNES internal document CO-SP-D0-EA-16222-CN v4.0.
Equivalent methods are available in the C++ and Java API: see **DorisFile** Class.
(this feature has been requested by the Sentinel-3 project).

3.3 Orbit Library

- **Orbit initialization functions extended: support for Sentinel-3 Doris files**
It is now possible to initialize an orbit_id with Doris files for Sentinel-3 using either **xo_orbit_init_file** or **xo_orbit_id_init.** Since a Doris file for Sentinel-3 does not contain any orbit

numbering information, the orbit number starts at 1 with the first Orbit State Vector (OSV) and increments at each ANX. The user can change this numbering using one of the following methods:

- After orbit_id initialization: by calling **xo_orbit_id_change** (see below);
- At orbit initialization with **xo_orbit_init_file** or **xo_orbit_id_init**:
  by using XO_ORBIT_INIT_DORIS_ORBNUM_ADJ_MODE (see below).

Equivalent methods are available in the C++ and Java API: see **OrbitId** Class.
(this feature has been requested by the Sentinel-3 project).

- **Orbit initialization functions extended: support for SP3 files**
  It is now possible to initialize an orbit_id with SP3 files using **xo_orbit_init_file** or **xo_orbit_id_init**.
  Equivalent methods are available in the C++ and Java API: see **OrbitId** Class.
  (this feature has been requested by the Metop-SG project).

- **Methods to change orbit numbers in an orbit_id**
  It is now possible to change the orbit numbering either by using a reference Orbit Scenario File (OSF) or setting a given orbit number at a given time.
  This can be achieved via:
    - The new function **xo_orbit_id_change;**
    - Using xo_orbit_init_file or xo_orbit_id_init functions and new modes **XO_ORBIT_INIT_(POF/ROF/DORIS)_ORBNUM_ADJ_MODE** (with xo_orbit_init_file, only change with OSF is alllowed).

  Equivalent methods are available in the C++ and Java API: see **OrbitId** Class.

- **xo_orbit_info function extended to compute Spacecraft Midnight (SMX)**
  SMX is the time just halfway the nadir day → night transition and the nadir night → day transition. Such transitions are times at which the Sun Zenith Angle (SZA, angle satellite- nadir-sun) is 90 deg. In the day → night transition, the SZA is increasing (i.e. there is a transition from SZA<90 to SZA>90). In the night → day transition the SZA is decreasing (i.e. there is a transition from SZA>90 to SZA<90). If more than one SMX is found in one orbit, only the one that is earlier in the orbit is returned. Note that this computation is disabled by default. The user must enable this computation by prior calling **xo_orbit_info_configure** (see below).
  Equivalent methods are available in the C++ and Java API: see **OrbitId** Class.
  (this feature has been requested by the Sentinel-5P project).

- **New function for xo_orbit_info output configuration: xo_orbit_info_configure**
  Equivalent methods are available in the C++ and Java API: see **OrbitId** Class.

3.4 Visibility Library

- **New function for zone coverage computation: xv_zonevistime_coverage**
  The function computes the portion of the input zone that is covered by a swath during a set of input time visibility intervals. The function computes:
    - The total coverage;
    - The coverage per interval;
    - The coverage per number of intervals;

- The cumulative coverage;
- The surface area of the zone.

Equivalent methods are available in the C++ and Java API: see **SwathId** Class.
(this feature has been requested by the Sentinel-2 project).

4 SOLVED PROBLEMS

The following Anomalies have been solved:

ANR Id	Description
525	xl_time_ascii_to_processing: Seconds are out of range error with leap second
535	extra zero in min/nominal/max values for the semi-major axis of GEO satellites
536	xv_sc_vis_time does not calculate correctly the stop times
538	xv_time_segments_delta does not check if start time > stop time for output segments
543	XP_LOS_surface_inter: No solutions found when intersection with DEM is computed *This problem was reported by the Sentinel-3 project.*
544	xp_dem_init: crash with long filename *This problem was reported by the Sentinel-3 project.*
545	Java ANXTime with TLE files does not converge
547	xv_zone_vis_time: memory allocation problem
549	xl_time_processing_to_ascii: problem with leap second handling
551	Outlier filter for OSVs removes the wrong OSV
555	xp_target_list_inter: Error returned instead of Warning
556	xp_target_list_inter: variable accessed inconsistently by threads
557	Visibility Lib, function xv_zone_vis_time: crash when zone segment is on the equator
559	When IERS Bulletin B is used and a leap second is detected, TAI-UTC is set to a wrong value
560	Memory leak in xf_tree_read
563	xv_zonevistime_compute: possible memory leak
564	xd_write_orbit_file: quality field cannot be written if quality idx > 8
566	xf_tree_cleanup_parser: global variable global_current_status accessed without locking the access for other threads
567	xo_orbit_init_file_precise: propag_flag wrongly set

5 KNOWN PROBLEMS

The updated list of known issues that will be resolved in a future relase can be found at the following URL:

http://eop-cfi.esa.int/index.php/mission-cfi-software/eocfi-software/branch-4-x/known-issues-branch-4

6 USER SUPPORT

For any question related to the usage of the EOCFI or to report a problem, please contact:

EOCFI Software Support Team

email: cfi@eopp.esa.int