EARTH OBSERVATION MISSION CFI SOFTWARE

Release Notes –Version 4.10

1 INTRODUCTION

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

2 USER SUPPORT

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

EOCFI Software Support Team

e-mail: cfi@eopp.esa.int

3 RELEASE DESCRIPTION

3.1 Software

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

Library Name	Version	Issue Date
File Handling	4.10	29 October 2015
Data Handling	4.10	29 October 2015
Lib	4.10	29 October 2015
Orbit	4.10	29 October 2015
Pointing	4.10	29 October 2015
Visibility	4.10	29 October 2015
EECommon (*)	4.10	29 October 2015

(*) 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

3.2 Documentation

The following documents are available:

Type	Document Name	Version
General	Mission Conventions Document	4.10
General	General Software User Manual	4.10
C API	Quick Start Guide	4.10
C API	File Handling Software User Manual	4.10
C API	Data Handling Software User Manual	4.10
C API	Lib Software User Manual	4.10
C API	Orbit Software User Manual	4.10
C API	Pointing Software User Manual	4.10
C API	Visibility Software User Manual	4.10

The documentation is available for download (and on-line browsing for C++ and JAVA APIs) 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

3.3 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 7 Operating Systems.
  - Software Requirements: Microsoft Visual C++ Compiler (Visual Studio 2010 Express or Professional edition)

- **MACIN64**
  - MACOSX on Intel (64-bits)
  - Platform Requirements: x86_64 based Mac Computer, Mac OS X version 10.10.x (Yosemite)
  - Software Requirements: Xcode 5.1 with clang compiler front end (gcc is an alias for clang)

  NOTE for MACIN64 platform, Xcode 5.x users:
  As of version 5 of Xcode the default compiler is clang (see http://clang.llvm.org/). clang is a compiler front end for C and C++ and can build an application linking against the EOCFI C / C++ libraries.
  The gcc anf g++ program provided within Xcode are aliases for clang.
  OpenMP is not supported in clang. Therefore, the –fopenmp switch shall not be used. Functions using parallelized computations, e.g. xp_target_list... functions will operate in single-threading mode.

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 (*)
  (in MACIN64, g++ is an alias for clang) and g++ compiler version 4.5.x for LINUX64 (*)
- Microsoft Visual C++ Compiler (Visual Studio 2010 Express or Professional edition) 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

(*) According to gcc documentation, forward compatibility is ensured up to gcc/g++ version 4.9.x.

3.4 Installation Packages

The CFI libraries are provided as zip packages:

API	Package Name	MD5 Checksum
C	EOCFI-4.10-CLIB-LINUX32_LEGACY.zip	3eea1d01236a59797adf6ca275b1cb0f
C	EOCFI-4.10-CLIB-LINUX64.zip	a970ef3ca65b92e5e766b2c40c37b660
C	EOCFI-4.10-CLIB-LINUX64_LEGACY.zip	6157eb9b93b6654967183a767800cb0d
C	EOCFI-4.10-CLIB-MACIN64.zip	1fc39ef5c34fbbaf3eb252bef155405d
C	EOCFI-4.10-CLIB-WINDOWS.zip	2d2db3932676afee66b185d2857c7b76
C++	EOCFI-4.10-CPPLIB-LINUX32_LEGACY.zip	48b40aa842219b2ebdbaf2fd79594e04
C++	EOCFI-4.10-CPPLIB-LINUX64.zip	1e304b71a6319197bb887ed88d526eba
C++	EOCFI-4.10-CPPLIB-LINUX64_LEGACY.zip	ec861986df053d914e22daf32e6c8560
C++	EOCFI-4.10-CPPLIB-MACIN64.zip	469091a5eaeb1edca44cbd1a8335b01c
C++	EOCFI-4.10-CPPLIB-WINDOWS_DLL.zip	a668eba688a42c04087fbf2cf5510c89
C++	EOCFI-4.10-CPPLIB-WINDOWS_STA.zip	86486e7ee89eb6e44bd9775a53396e65
JAVA	EOCFI-4.10-JAVALIB-LINUX32_LEGACY.zip	6b3d80bf93ef71123de3abb60bae3d43
JAVA	EOCFI-4.10-JAVALIB-LINUX64.zip	01cbdbc8c5048dc37387b258ba2de987
JAVA	EOCFI-4.10-JAVALIB-LINUX64_LEGACY.zip	3bd1b0b33a6490827a253de3dd33a8c5
JAVA	EOCFI-4.10-JAVALIB-MACIN64.zip	0f41d23d3d86114b04e797795e7d0dc3
JAVA	EOCFI-4.10-JAVALIB-WINDOWS.zip	e133c4ad93ddaff39aabaa725bca7b7c

(*) Dynamic libraries (DLLs)

(**) Static libraries

Information on how to get and use the supported DEM datasets can be found at the following URL:

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

3.5 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;
- Section 6 “LIBRARY USAGE” of each Library User Manual.

The xerces library and SAX2Count application are no longer required by the XML validation function and tool in the Data Handling library.

The EOCFI libraries make use of the following third-party libraries:

- pthreads (POSIX threads): this library is normally pre-installed in Linux and Mac OS X systems. For Windows, the library is provided in the cfi_tools directory within the distribution package.

Pthreads is covered by the GNU Lesser General Public License.
(see https://www.sourceware.org/pthreads-win32/copying.html).

- libxml2 (see http://xmlsoft.org/): for reading and writing XML files.
- libgeotiff (see http://trac.osgeo.org/geotiff/), libtiff (see http://www.libtiff.org/), libproj (see http://trac.osgeo.org/proj/): for reading ASTER GDEM files.

Please refer to Section 6 of the User Manual of each Library for specific usage instructions. Terms and conditions for usage of such libraries is detailed in the text file (included in the distribution package) TERMS_AND_CONDITIONS.TXT.

libxml2, libgeotiff, libtiff and libproj are provided:

- in the C API distribution packages: as separated static libraries (see Section 6 of each User Manual for instruction on how to link them to the application program).
- in the C++ / Java APIs distribution packages: as separated dynamic libraries (see Section 6 of each User Manual for instruction on how to link them to the application program). In the Java API for MAC OS X platform, due to incompatibilities with system libraries, they are instead embedded in the EOCFI libraries.

User applications using the Pointing library need to be built with OpenMP support (adding –fopenmp switch in gcc or /fopenmp in Visual Studio, see Section 6 of the Pointing User Manual).

In Mac OS X platforms, OpenMP is not supported in clang, therefore no additional switch is required. Functions will operate in single-threading mode.

In WINDOWS platforms, OpenMP is not supported in Visual Studio 2010 Express, therefore no additional switch is required. Functions will operate in single-threading mode.

The XML validation function and tool in the Data Handling library use the libxml2 library. For WINDOWS platform, it is required to link the user application against the ws2_32.lib.

4 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.

- **Run-time improvements in target functions**
  - **Pointing Library:** target functions have been improved, so that now the run-time is faster w.r.t. previous versions, as shown by the following table (test executed on three target methods from the C++ API):

Target Method	Run-time with v4.9.1 [ms]	Run-time with v4.10 [ms]	Speed-up (*)
targetInter	0.010593	0.002236	4.74
targetGeneric	0.007277	0.001429	5.09
targetRange	0.017279	0.008033	2.15

(*) the speed-up is the ratio between execution with v4.9.1 and v4.10.

- **Light Propagation Model**
  - **Lib Library:** a new model type XL_MODEL_TYPE_LIGHT_PROPAGATION is available. This model type is disabled by default (see section 7.49 of Lib Software User Manual).
  - **Pointing Library:** When the light propagation model is enabled, the target functions keep into account the time spent by a generic signal travelling at the speed of light to:
    - in the TRANSMITTER mode: go from the satellite to the target;
    - in the RECEIVER mode: go from the target to the satellite.

    (see section 4.1.2.3 of the Pointing Software User Manual)

    NOTE: due to the additional computations, the execution run-time is longer when the model is activated.

- **Improved error handling and diagnostics for Doris and Orbit files**
  - **Data Handling Library:**
    - Additional checks in **xd_read_doris** (see section 7.7 of the Data Handling Software User Manual):
      - packet is discarded and a warning is raised if at least one of the following conditions is detected: CRC error; quality field = 0xFFFFFFFF (packet not valid); OSV time going back or repeated.
      - A warning is raised and the packet is not discarded if at least one of the following conditions is detected: sequence counter not increasing (except for counter wrapping); gap between two consecutive OSVs.
    - Additional checks in **xd_read_orbit_file** (see section 7.5 of the Data Handling Software User Manual):
      - A warning is raised if at least one of the following conditions is detected: OSV with time going back or duplicated; gap found.
    - New function, **xd_orbit_file_diagnostics** (see section 7.61 of the Data Handling Software User Manual): it computes diagnostics data related to an orbit file. Such data can be analysed to detect problems in the file or identify fragments of the file to be discarded.
  - **Orbit Library:** orbit intialisation using files with list of OSVs (**xo_orbit_init_file, xo_orbit_id_init**, see sections 7.5 and 7.7 of the Orbit Software User Manual):
    - OSVs with time going back or duplicated are discarded (warning is raised at 1st occurence);
    - When a gap that does not allow interpolation is detected, the function stops loading OSVs and a warning is raised;
    - Inconsistent orbit numbers are adjusted (warning is raised at 1st occurrence).

- **Improved initialisation of Time Id / Orbit Id with data structures**
  - **Lib Library:** in **xl_time_id_init** (see section 7.3 of the Lib Software User Manual) it is now possible to set orbit_file_mode to XO_TIMEMOD_AUTO (it is no longer required to specify the type of data).
  - **Orbit Library:** in **xo_orbit_id_init** (see section 7.7 of the Orbit Software User Manual) it is now possible to set orbit_file_mode to XO_ORBIT_INIT_AUTO (it is no longer required to specify the type of data).

- **Improved logging of errors / messages raised by libxml2 functions**
  - **All Libraries:** in previous versions, errors / messages raised by libxml2 functions were logged on standard output/error. From this version, by default the logging on standard output/error is deactivated. It can be activated/deactivated by using new functions **xf_verbose/xf_silent** (**File Handling Library**).

- **Support for new DEM type: ACE2 30 arcsec**
  - **Data Handling Library:** New DEM Configuration File format (the new value ACE2_30SEC is allowed within the `<Dataset_Model>` tag).
  - **Pointing Library:** Pointing functions have been adapted to support ACE2 30 arcsec DEM (see http://tethys.eaprs.cse.dmu.ac.uk/ACE2/shared/overview)

- **Improvement in functions to read SP3 and OEM files**
  - **Data Handling Library:** xd_read_sp3 and xd_read_oem have been improved. It is now possible to extend the selection of OSVs by providing either the number of additional OSVs or a time interval.

    NOTE: the function interfaces have been changed, therefore user applications using these functions need to be re-compiled.

5 SOLVED PROBLEMS

The following Anomalies have been solved:

ANR Id	Description
548	SDF schemas: 'count' attribute of List_of_Harmonics_Type should be allowed to be zero (*)
561	xp_change_frame: error calling xl_change_cart_cs when satellite attitude initialised with file
573	Problem in xo_orbit_to_time: when sec=msec=0 the correct ANX time is not returned.
580	xo_orbit_info: SMX not computed correctly in some specific cases.
581	xv_orbit_extra: error returned when calculating Sun Zenith Angle.
582	xd_read_doris: wrong error messages, e.g. could not find keyword: SENSING_START=”
616	xv_zone_vis_time: error with wide swath and zone covering latitudes near the poles.
617	xo_orbit_id_init: double free in error handling causes crash. *(this issue has been reported by the Sentinel-1A IPF Team)*
618	xv_zonevistime_coverage: problem when last zone point is equal to first. *(this issue has been reported by the Sentinel-2A Mission Planning Team)*
635	xp_target_inter: Intersection with DEM cannot be computed. *(this issue has been reported by the Sentinel-3A IPF Team)*
637	Absolute Orbit = 0 in OSVs shall be accepted by XML validation (*)

(*) these ANRs have been resolved by issuing a new version of the relevant schemas

6 PROBLEMS

The updated list of known issues that will be resolved in a future release 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