EARTH OBSERVATION MISSION CFI SOFTWARE

Release Notes –Version 4.13

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	4.13	12 April 2017
Data Handling	4.13	12 April 2017
Lib	4.13	12 April 2017
Orbit	4.13	12 April 2017
Pointing	4.13	12 April 2017
Visibility	4.13	12 April 2017
EECommon (*)	4.13	12 April 2017

(*) 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	4.13
General	General Software User Manual	4.13
C API	Quick Start Guide	4.13
C API	File Handling Software User Manual	4.13
C API	Data Handling Software User Manual	4.13
C API	Lib Software User Manual	4.13
C API	Orbit Software User Manual	4.13
C API	Pointing Software User Manual	4.13
C API	Visibility Software User Manual	4.13

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-bit (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-bit (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 (*)

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

Note that LINUX32_LEGACY and LINUX64_LEGACY above will be discontinued in 2018

- **LINUX64**
  - LINUX 64-bit
  - 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 (*)

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

- **WINDOWS32**
  - Microsoft WINDOWS PC (32-bit)
  - Platform Requirements: x86 based PC, Microsoft Windows 7 Operating Systems.
  - Software Requirements: Microsoft Visual C++ Compiler (Visual Studio 2010 Express or Professional edition, 32-bit)

- **WINDOWS64**
  - Microsoft WINDOWS PC (64-bit)
  - Platform Requirements: x86_64 based PC, Microsoft Windows 7 Operating Systems.
  - Software Requirements: Microsoft Visual C++ Compiler (Visual Studio 2010 Express or Professional edition, 64-bit)

- **MACIN64**
  - MACOSX on Intel (64-bit)
  - 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 and 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 8 for all platforms

3.4 Installation Packages

The CFI libraries are provided as zip packages:

API	Package Name	MD5 Checksum
C	EOCFI-4.13-CLIB-LINUX32_LEGACY.zip	f5e1c03100a7589810574fa1bed11cde
C	EOCFI-4.13-CLIB-LINUX64.zip	806764e237f2eef5d8563bf3e11986bc
C	EOCFI-4.13-CLIB-LINUX64_LEGACY.zip	236d1fa4de7aa5fdc6f16952e59cbf14
C	EOCFI-4.13-CLIB-MACIN64.zip	b20572129bd9b6c9306205e81decf3b2
C	EOCFI-4.13-CLIB-WINDOWS32.zip	b78ed758c19e11f0685e45e95a767cf4
C	EOCFI-4.13-CLIB-WINDOWS64.zip	bf0be5edd76f81073d5e44ee5487d78b
C++	EOCFI-4.13-CPPLIB-LINUX32_LEGACY.zip	a4d486fce671119fe7b5842694d39ee1
C++	EOCFI-4.13-CPPLIB-LINUX64.zip	080fd2fcaf5a07f62e6e3f0a746698f1
C++	EOCFI-4.13-CPPLIB-LINUX64_LEGACY.zip	ace8439874627cb4d3a1b21545c86519
C++	EOCFI-4.13-CPPLIB-MACIN64.zip	1e9c2ae3fca37245009d1b7c62291e28
C++	EOCFI-4.13-CPPLIB-WINDOWS32_DLL.zip	31dad671951dc35a17f168971d7b8096
C++	EOCFI-4.13-CPPLIB-WINDOWS32_STA.zip	b5bcdcd94ea10bc391444aa32ef31d5e
C++	EOCFI-4.13-CPPLIB-WINDOWS64_DLL.zip	8aa78fa7c5854beac096bb78287f9800
C++	EOCFI-4.13-CPPLIB-WINDOWS64_STA.zip	130eebe2346410b072533a6144b0c057
JAVA	EOCFI-4.13-JAVALIB-LINUX32_LEGACY.zip	e91e9a5b7ff5d907b77b8babd9a41f23
JAVA	EOCFI-4.13-JAVALIB-LINUX64.zip	a33f6c22c3a5896027f3f5933edfdc6a
JAVA	EOCFI-4.13-JAVALIB-LINUX64_LEGACY.zip	aa685c65b924823b297200fb2e1b67e7
JAVA	EOCFI-4.13-JAVALIB-MACIN64.zip	4e8ee5fc879ef0c36a92faea59507d2a
JAVA	EOCFI-4.13-JAVALIB-WINDOWS32.zip	471735fed93387fd5aa87465572a2692
JAVA	EOCFI-4.13-JAVALIB-WINDOWS64.zip	a63ad41d79078890010e8c8e184ee236

(*) 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 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 are 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, see Section 6 of the Pointing User Manual).

OpenMP is not supported in clang (Mac OS X) and Visual Studio (Windows), therefore no additional switch is required. Functions will operate in single-threading mode.

The XML validation function and tool in the Data Handling library uses the libxml2 library. For Windows platforms, 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.

- **TLE designator updated for Sentinel-2B**

- **Support for Earth Observation Ground Segment File Format Standard version 3.**

5 SOLVED PROBLEMS

The following Anomalies have been solved:

ANR Id	Description
657	Wrong cycle number (increased or decreased) when appending orbital change with longitude jump
662	OrbitExtra calculation with TLE: some parameters (e.g. MLST) are not computed properly
680	Attitude compute method does not check OSV Co-ordinate system and assumes EF (C++,JAVA)
682	Java API: targetListInter memory leak

686	Warning is not raised when attitude is computed and quaternions are extrapolated
688	Problems in gen_rof: does not check step > 0; with orbit range as input, 1st OSV has Z < 0
691	Attitude::getAttitudeData() does not properly calculate RPY with quaternions initialisation

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