EARTH OBSERVATION MISSION CFI SOFTWARE

Release Notes –Version 4.12

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	4.12	11 November 2016
Data Handling	4.12	11 November 2016
Lib	4.12	11 November 2016
Orbit	4.12	11 November 2016
Pointing	4.12	11 November 2016
Visibility	4.12	11 November 2016
EECommon (*)	4.12	11 November 2016

(*) 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	4.12
General	General Software User Manual	4.12
C API	Quick Start Guide	4.12
C API	File Handling Software User Manual	4.12
C API	Data Handling Software User Manual	4.12
C API	Lib Software User Manual	4.12
C API	Orbit Software User Manual	4.12
C API	Pointing Software User Manual	4.12
C API	Visibility Software User Manual	4.12

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-4.12-CLIB-LINUX32_LEGACY.zip	096005e39f33b22b4490391638624b54
C	EOCFI-4.12-CLIB-LINUX64_LEGACY.zip	867b4f8199ff0873641c8b86c2c13d34
C	EOCFI-4.12-CLIB-LINUX64.zip	c2fbfd7b5e15d7bc71fed81b56d7eaf4
C	EOCFI-4.12-CLIB-MACIN64.zip	5d46390a663a23cb791a49f0b5b7d150
C	EOCFI-4.12-CLIB-WINDOWS32.zip	8164205e6ac45c83aea4e553f438666b
C	EOCFI-4.12-CLIB-WINDOWS64.zip	7f9dd03b37982fe220a3713ac3c3538a
C++	EOCFI-4.12-CPPLIB-LINUX32_LEGACY.zip	78d4e0b788e35dbc64df25f1e84c7013
C++	EOCFI-4.12-CPPLIB-LINUX64_LEGACY.zip	233e55747b99a099bb86eab9c06fbcf5
C++	EOCFI-4.12-CPPLIB-LINUX64.zip	81b408c0a0797047197ef0d564b79a33
C++	EOCFI-4.12-CPPLIB-MACIN64.zip	35ec5d3708d5e70acf2285ad2f9a2880
C++	EOCFI-4.12-CPPLIB-WINDOWS32_DLL.zip	bd403458e2eee3253d52a8b583b690e9
C++	EOCFI-4.12-CPPLIB-WINDOWS32_STA.zip	4e71f3801e9359f3dfc4e176befbd6fa
C++	EOCFI-4.12-CPPLIB-WINDOWS64_DLL.zip	c7c2f51e081ec0bc51726854aa7609ee
C++	EOCFI-4.12-CPPLIB-WINDOWS64_STA.zip	c091c776b39be567f16bf051cd4c3997
JAVA	EOCFI-4.12-JAVALIB-LINUX32_LEGACY.zip	b94670c3fd153465c5b2618afba00231
JAVA	EOCFI-4.12-JAVALIB-LINUX64_LEGACY.zip	4f997e45287d022136ad0498838976a7
JAVA	EOCFI-4.12-JAVALIB-LINUX64.zip	a02a10ba08dd693e225547255d29d660
JAVA	EOCFI-4.12-JAVALIB-MACIN64.zip	3af63792e68e7d4101669eb3fa86d70d
JAVA	EOCFI-4.12-JAVALIB-WINDOWS32.zip	13b2b1d3b61552997477242169cdb845
JAVA	EOCFI-4.12-JAVALIB-WINDOWS64.zip	bad0f8f7c73f1e1f0c9b2d4b2cf79471

(*) 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.

- **xl_time_ascii_to_ascii function: 9999-12-31T23:59:59 (and similar) is now accepted as End of Mission string**

- **Update of Attitude Definition File format**
  Quaternion_Plus_Angle and *Quaternion_Plus_Matrix* initialisation modes: quaternions are provided via an external file. For example:

```
<Quaternion_Plus_Matrix>
    <Matrix_Model>
        …
    </Matrix_Model>
    <Quaternion_File> quaternions_att_file.xml </Quaternion_File>
</Quaternion_Plus_Matrix>
```

- **TLE designator updated for Sentinel-1B**

- **Improments in run-time of xp_target_list_inter function**
  xp_target_list_inter is now about 4 times faster than xp_target_inter.
  For example, on LINUX64 (Intel Xeon E5-2620 2.00GHz, 64GB RAM) calculating 1000000 geolocations requires:
  - 2.94 seconds with xp_target_inter;
  - 0.75 seconds with xp_target_list_inter.

- **Improments in run-time of DEM intersection algortihm**
  When the DEM is initialised in PRELOAD mode, a dedicated algorithm is used to compute the intersection of the Line of Sight with the DEM. This algorithm is much faster than the one used in FIFO mode. The following table shows the time required to compute on LINUX64 (Intel Xeon E5-2620 2.00GHz, 64GB RAM) the intersection of 1000000 points using two different DEM types, GETASSE30 and DEM ACE2 9 arcsec with geoid computation disabled. Runtimes with ACE2 are longer due to its higher resolution.

DEM Type	run-time FIFO mode [s]	run-time PRELOAD mode [s]
GETASSE30	10.0	2.8
ACE2	34.0	8.5

  Note: the PRELOAD mode can be set in the DEM configuration file by changing the Cache Type to PRELOAD_CACHE. Before running the geolocations, the required DEM area must be loaded in memory using function xp_dem_id_configure (see Pointing SUM).
  Please contact the EOCFI Software team (cfi@eopp.esa.int) for specific support on this feature.

- **Improvements in Orbit State Vector Extrapolation**
  Error due to extrapolation is smaller than in previous versions. For example, with a list of OSVs at 10 sec time interval, extrapolation is possible up to 40 sec with an error of 7 m (in previous versions the error was about 50 m).

- **Extrapolation enabled in Attitude / Quaternions computations**

- **New file type: Field of View configuration**
  - **Data Handling Library:** the Field of View configuration file defines a field of view in terms of visibility masks. The new function `xd_read_fov_constraints_file` reads the file and gives as output a data structure that can be used as input for other functions, e.g. xv_sc_vis_time.

- **New function for Orbit State Vector (OSV) sanity check**
  - **Orbit Library:** the function `xo_osv_check` checks that the OSV is within tolerances defined for the input satellite nominal orbit.
    Note: this check is no longer done internally by EOCFI functions.

- **New function to set the time step used for S/C visibility computations**
  - **Visibility Library:** the `xv_set_sc_vis_time_step` function sets the time step used by the function xv_sc_vis_time. A shorter time step gives more accurate results but a longer run-time.

- **Support for Earth Observation Ground Segment File Format Standard version 3.**
  - **Data Handling Library:** new function `xd_set_file_format_standard_version` to enable File Format Standard version 3 for reading and writing.
  - **Orbit / Pointing Library:** file generator tools (e.g. gen_rof) have an additional switch "-eoffs" to enable a specific file format standard version. When "-eoffs 3" is used, output file is generated according to File Format Standard v3.

5 SOLVED PROBLEMS

The following Anomalies have been solved:

ANR Id	Description
647	targetReflected method: unexpected fail. *(in some sporadic cases the algorithm fails to find a solution. A degraded solution is now returned and a warning is raised).*
655	targetInter method: memory leak in Java API.
658	Error/Warnings reading Doris files for Sentinel-3 *(this problem has been reported by the Sentinel-3 IPF Team).*

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