

# EARTH OBSERVATION MISSION CFI SOFTWARE

## Release Notes –Version 4.14

### 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](mailto: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	4.14	21 November 2017
Data Handling	4.14	21 November 2017
Lib	4.14	21 November 2017
Orbit	4.14	21 November 2017
Pointing	4.14	21 November 2017
Visibility	4.14	21 November 2017
ECommon (*)	4.14	21 November 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	4.14
General	General Software User Manual	4.14
C API	Quick Start Guide	4.14
C API	File Handling Software User Manual	4.14
C API	Data Handling Software User Manual	4.14
C API	Lib Software User Manual	4.14
C API	Orbit Software User Manual	4.14
C API	Pointing Software User Manual	4.14
C API	Visibility Software User Manual	4.14

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.

- **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)

**Note that LINUX32\_LEGACY and WINDOWS32 platforms will be discontinued in 2018**

- **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 7.2 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-4.14-CLIB-LINUX32_LEGACY.zip	ca9732337dee2e5ebf1996a9cf92eb5f
C	EOCFI-4.14-CLIB-LINUX64.zip	b7fdc798e281cec9a0cec7dd36227a09
C	EOCFI-4.14-CLIB-LINUX64_LEGACY.zip	843ed147c22d1f1cfd6c6f99ed7b65ec
C	EOCFI-4.14-CLIB-MACIN64.zip	981b0d4601794ff5d3b2cd8c81b37773
C	EOCFI-4.14-CLIB-WINDOWS32.zip	df633480fbd8da4fb08ae574711dac9d
C	EOCFI-4.14-CLIB-WINDOWS64.zip	60351ee8e6b764592d08634ea97bf9cb
C++	EOCFI-4.14-CPPLIB-LINUX32_LEGACY.zip	fe3115afd9e369432dbebac20163938d
C++	EOCFI-4.14-CPPLIB-LINUX64.zip	a3bb50098818e831fc349ec7bf3ce4b3
C++	EOCFI-4.14-CPPLIB-LINUX64_LEGACY.zip	e2c21d427b8c56c84c0dd94e80bb5851
C++	EOCFI-4.14-CPPLIB-MACIN64.zip	4a2eed604e0b9e939a068e2cd4e7d731
C++	EOCFI-4.14-CPPLIB-WINDOWS32_DLL.zip	6fbedf87c04691bc37b66c6b66757262
C++	EOCFI-4.14-CPPLIB-WINDOWS32_STA.zip	4db9680d136e8f83e1b6acc3fd9a62d9
C++	EOCFI-4.14-CPPLIB-WINDOWS64_DLL.zip	fa9e986a18535362657791e3fef4a624
C++	EOCFI-4.14-CPPLIB-WINDOWS64_STA.zip	c4ec8e76128d5d2d7199ce426af23b6c
JAVA	EOCFI-4.14-JAVALIB-LINUX32_LEGACY.zip	871fd8a86436696d9c93bdb54fbfb80b
JAVA	EOCFI-4.14-JAVALIB-LINUX64.zip	f77fc33ee898956fc6958d35cff6343e
JAVA	EOCFI-4.14-JAVALIB-LINUX64_LEGACY.zip	e7ab3d9cd50e3e4087ce4abf033c69a7
JAVA	EOCFI-4.14-JAVALIB-MACIN64.zip	55ba3b60f71887c7b94fed8710383954
JAVA	EOCFI-4.14-JAVALIB-WINDOWS32.zip	41331e7011bb1991c67cc46bf297de20
JAVA	EOCFI-4.14-JAVALIB-WINDOWS64.zip	e3e644ce11a275f669857893031617c6

(\*) 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.gnu.org/licenses/lgpl-3.0.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.

- **Support for new mission: FLEX**  
*Feature requested by the FLEX project team*
- **Tools and functions for generation of orbit and attitude file: TLE file is allowed as input.**  
`xo_gen_pof / xo_gen_rof` functions, `gen_pof / gen_rof` command line tools (Orbit library) and `xp_gen_attitude_file` function, `gen_attitude` command line tool (Pointing library) accept also TLE file as input for the calculation of orbit data.
- **Orbit propagation (e.g. `xo_osv_compute` function): when the orbit Id is initialised with a TLE file it is allowed to propagate after 1 day of the TLE epoch.**
- **Visibility library, `xv_zonevistime_compute` function: visibility segments can be calculated also with a ROF (Restituted Orbit File) covering less than one orbit.**
- **Visibility functions: visibility segments can be computed with a new instrument swath geometry defined by azimuth and incidence angle.**

The Swath Definition File format has been extended to support a new geometry type: Incidence angle.

```
<Swath_Point>
  <Incidence_Angle_Geometry>
    <Azimuth unit="deg">+090.000000</Azimuth>
    <Incidence_Angle unit="deg">+20.000000</Incidence_Angle>
    <Altitude unit="m">+0.000</Altitude>
  </Incidence_Angle_Geometry>
</Swath_Point>
```

- New attitude model: MetOp-SG**  
 The Z axis of the attitude frame is computed with an approximation for the local normal vector (see section 7.2.3.5 of the Pointing SUM and section 7.2 of the Mission Conventions Document, Mission Specific customisations).  
*Feature requested by EPS-SG INR team (EUMETSAT)*
- New functions to convert CCSDS Unsegmented Time Code (CUC) to processing time (and vice-versa)**  
 See Lib SUM, sections 7.19 ([xl\\_time\\_cuc\\_to\\_processing](#)) and 7.20 ([xl\\_time\\_processing\\_to\\_cuc](#)).  
*Feature requested by EPS-SG INR team (EUMETSAT)*
- Support for Jason-CS Doris files**  
 Jason-CS Doris files can be used to initialise an orbit Id (e.g. using `xo_orbit_init_file` function in the Orbit library) and can be ingested by `xd_read_doris` function in Data Handling library.  
*Feature requested by Jason-CS project team*
- Function `xv_gen_scf` (Visibility library) generates SCF files compliant with ESOV supported format**
- New function `xp_free_target_id_data` (Pointing library) to release data of `xp_target_id_data` type**

## 5 SOLVED PROBLEMS

The following Anomalies have been solved:

ANR Id	Description
656	Zone visibility function does not calculate all segments when using TLE file <i>Problem reported by the ESOV development team</i>
684	ANX MLST calculated by the <code>xo_orbit_info</code> is not correct when an Orbit Scenario File with MLST drift different from zero is used
703	<code>targetListInter</code> crash when no Intersection between LOS vector and Earth Surface <i>Problem reported by the EPS-SG INR team (EUMETSAT)</i>
704	<code>xo_orbit_init_def</code> : wrong validity time <i>Problem reported by the ESOV development team</i>
708	<code>targetListInter</code> : exception when no solution is found

	<i>Problem reported by the Sentinel-5 UVNS L1bPP development team</i>
709	C++ / JAVA: getGeodetic assumes input cartesian co-ordinates in EF <i>Problem reported by the MetOp-SG MetImage project industrial team</i>
714	xp_target_range: when two solutions are found, only one is returned
715	xo_osv_compute: error close (before) to ANX when MLST drift is used <i>Problem reported by Sentinel-6 L1b Simulator development team</i>
718	Error returned by xv_zone_vis_time with POINT & CIRCLE zones <i>Problem reported by the ESOV development team</i>

## 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>