

# EARTH OBSERVATION MISSION CFI SOFTWARE

## Release Notes –Version 4.8

### 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**

**email: [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	4.8	29 October 2014
Data Handling	4.8	29 October 2014
Lib	4.8	29 October 2014
Orbit	4.8	29 October 2014
Pointing	4.8	29 October 2014
Visibility	4.8	29 October 2014
EECommon (*)	4.8	29 October 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>

## 3.2 Documentation

The following documents are available:

Type	Document Name	Version
General	Mission Conventions Document	4.8
General	General Software User Manual	4.8
C API	Quick Start Guide	4.8
C API	File Handling Software User Manual	4.8
C API	Data Handling Software User Manual	4.8
C API	Lib Software User Manual	4.8
C API	Orbit Software User Manual	4.8
C API	Pointing Software User Manual	4.8
C API	Visibility Software User Manual	4.8

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 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 10.9.x (Mavericks)
- 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)
- 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

## 3.4 Installation Packages

The CFI libraries are provided as zip packages:

API	Package Name	MD5 Checksum
C	EOCFI-4.8-CLIB-LINUX32_LEGACY.zip	27d9ccd1c52897b42d0940f7c6a38b98
C	EOCFI-4.8-CLIB-LINUX64.zip	33eda6f8b9d1b04af0d9054a5eec086a
C	EOCFI-4.8-CLIB-LINUX64_LEGACY.zip	408ce3c643dff32026782f3e80922161
C	EOCFI-4.8-CLIB-MACIN64.zip	90008396f4307b773e38e6bc28deae67
C	EOCFI-4.8-CLIB-WINDOWS.zip	ae813837b3f9e473c7086a3b4fbb55d0
C++	EOCFI-4.8-CPPLIB-LINUX32_LEGACY.zip	e9b3f8e25adf7d0eafe8e607a3921632
C++	EOCFI-4.8-CPPLIB-LINUX64.zip	1d22f07c8e1690cc3c9a6fa6e4a0182f
C++	EOCFI-4.8-CPPLIB-LINUX64_LEGACY.zip	3db23de1b0a60b2e75c1f23784d71d04
C++	EOCFI-4.8-CPPLIB-MACIN64.zip	3af00c8ce588b545977b9063d768d0f9
C++	EOCFI-4.8-CPPLIB-WINDOWS_DLL.zip (*)	5d0b9fe8cf6f82d8ca9e90db6cd7306e
C++	EOCFI-4.8-CPPLIB-WINDOWS_STA.zip (**)	945e60e9934da0366f5e38d6d880f82b
JAVA	EOCFI-4.8-JAVALIB-LINUX32_LEGACY.zip	c7f6c5b182b785e4f447828c6436cc7a
JAVA	EOCFI-4.8-JAVALIB-LINUX64.zip	c62177f7cbcfce0ea2fdd58322c7f0fb1
JAVA	EOCFI-4.8-JAVALIB-LINUX64_LEGACY.zip	7b1566c1ec0280d289a4c4a8f002fdc9
JAVA	EOCFI-4.8-JAVALIB-MACIN64.zip	bcd81f927c518ac75c8d60dee787f24a
JAVA	EOCFI-4.8-JAVALIB-WINDOWS.zip	28c636d1b652e76b6e5b601ea684862a

(\*) 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>

ASTER GDEM is available for download here: <http://asterweb.jpl.nasa.gov/gdem.asp>

## 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 required by the XML validation function and tool in the Data Handling library. See Section 6 of the Data Handling 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 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 static libraries as separated libraries distribution package (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: 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.

## 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.

### 4.1 Data Handling Library

- **New Attitude File format and Attitude Definition File format**  
The `<Inertial_Ref_Frame>` has been replaced with `<Reference_Frame>`. Reading functions `xd_read_att` and `xd_read_att_def` can read files with any of the two tags. Writing functions `xd_write_att` and `xd_write_att_def` generate files with only the new tag.
- **New DEM Configuration File format**  
Two new values (`GETASSE30_V3` and `GDEM_V2`) are allowed within the `<Dataset_Model>` tag.
- **`xd_read_dem` function extended to read new DEM types**  
This function can read also ASTER GDEM dataset files.
- **New function: `xd_xslt_add`**  
The `xd_xslt_add` function adds to the input XML file the `<xml-stylesheet>` tag with reference to the default style sheet (see also Section 7.58 of the Data Handling User Manual). Examples of style sheets can be found in the distribution package, in the directory `files/xslt`.  
Equivalent methods are available in the C++ and Java API: see `xsltAdd` method, `XsltUtil` Class.

## 4.2 Lib Library

- **New function for quaternion interpolation: [xl\\_quaternions\\_interpol](#)**  
This functions interpolates between two quaternions using the SLERP algorithm. Equivalent methods are available in the C++ and Java API: see [quaternionsInterpol](#) method in the LibFunc Class.
- **New Coordinate System: Launch Inertial Frame (LIF)**  
The LIF is a user defined inertial frame. The definition is given in the Mission Convention Document, Section 5.1. Conversion from and to this frame is not enabled by default. The user has to enable it and provide the reference parameters (i.e. longitude and UTC time) using function [xl\\_time\\_set\\_id\\_data](#). Further details can be found in the Lib User Manual, section 7.23.1 (overview of [xl\\_change\\_cart\\_cs](#) function).
- **Model Id: new Sun Model (XL\_MODEL\_SUN\_TRAVEL\_TIME) added**  
If this model is enabled, target and attitude computations (with functions in the Pointing library) will take into account Sun light travel time.

## 4.3 Orbit Library

- **File generation tools: new switch `-with_xslt`**  
Tools for file generation in XML format (e.g. [gen\\_pof](#)) provide an additional option to add to the input file the `<xml-stylesheet>` tag with reference to the default style sheet (see also Section 7.58 of the Data Handling User Manual). Examples of style sheets can be found in the distribution package, in the directory `files/xslt`.

## 4.4 Pointing Library

- **Support for new DEM types: GETASSE30 V3 and ASTER GDEM**  
Pointing functions have been adapted to support GETASSE30 V3 (see <http://earth.esa.int/services/amorgos/download/getasse/>) and ASTER GDEM (see <http://asterweb.jpl.nasa.gov/gdem.asp>).  
Note: due to the high resolution of the ASTER GDEM, loading files in memory using CACHE methods (see Pointing User Manual, Section 7.62.2) may be slow and is recommended only when a large number of accesses need to be done.
- **[xp\\_sat\\_nominal\\_att\\_init\\_harmonic](#), [xp\\_sat\\_att\\_init\\_harmonic](#), [xp\\_instr\\_att\\_init\\_harmonic](#) functions extended**  
It is now possible to initialize the attitude ids with harmonics and true latitude in Earth Fixed (angle type can be set to `XP_ANGLE_TYPE_TRUE_LAT_EF`).
- **Sun light travel time correction in target and attitude functions**  
When Sun position is relevant, a correction can be applied in order to compensate the travel time of Sun light. This correction is not applied with the default model. To activate this correction, the Sun model must be initialized with the `XL_MODEL_SUN_TRAVEL_TIME` value using the function [xl\\_model\\_init](#) (see Lib library User Manual).

- **New function `xp_attitude_transform`**

The `xp_attitude_transform` CFI function allows the user to change the reference frame in which the internal data of the attitude ids are expressed.

Remark: all attitude related computations are performed in the True of Date Coordinate System.

This means that, if one attitude id has been initialized with data expressed in another reference frame (e.g. attitude file with quaternions expressed in EF), at each call of a function using such attitude id (e.g. `xp_attitude_compute`) one or more conversions to True of Date will be performed. If the reference frame is changed to True of Date (using `xp_attitude_transform`), such conversions will not be executed and this will result in a run-time performance improvement.

Equivalent methods are available in the C++ and Java API: see **`attitudeTransform`** method in the `AttitudeDef` Class.

## 4.5 Visibility Library

- **`xv_sc_vis_time` : added flag in input structure to enable/disable the masks**

The structure of type `xv_az_el_mask` (used to define inclusive and exclusive masks) has now an additional field *status* to enable or disable the mask.

Note: this flag needs to be set explicitly by the user. Therefore, code using previous versions of EOCFI has to be modified accordingly (i.e. the field has to be set to the requested value).

## 5 SOLVED PROBLEMS

The following Anomalies have been solved:

ANR Id	Description
534	Call to <code>xp_target_sc</code> always returns warning message if <code>sat_id2</code> is a GEO
537	Sometimes <code>gen_osf_append_orbit_change</code> calculates incorrectly the relative orbit and/or cycle number
554	<code>xo_gen_tle</code> : validity encoded in filename is not compliant with file format standard
568	<code>xp_instr_att_matrix_init</code> : memory leak (this issue has been reported by the Sentinel-3 IPF Team)
575	<code>xp_target_range</code> : solution cannot be found in some specific conditions (this issue has been reported by the Sentinel-1 Project Team)
574	Errors using <code>xv_zonevistime_coverage</code> (this issue has been reported by the Sentinel-2 Mission Planning 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>