

EARTH OBSERVATION MISSION CFI SOFTWARE

Release Notes –Version 4.5

1 INTRODUCTION

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

2 RELEASE DESCRIPTION

2.1 Software

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

Library Name	Version	Issue Date
File Handling	4.5	4 March 2013
Data Handling	4.5	4 March 2013
Lib	4.5	4 March 2013
Orbit	4.5	4 March 2013
Pointing	4.5	4 March 2013
Visibility	4.5	4 March 2013
EECommon (*)	4.5	4 March 2013

(*) 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>

2.2 Documentation

The following documents are available:

Type	Document Name	Version
General	Mission Conventions Document	4.5
General	General Software User Manual	4.5
C API	Quick Start Guide	4.5
C, C++, Java APIs	File Handling Software User Manual	4.5
C, C++, Java APIs	Data Handling Software User Manual	4.5
C, C++, Java APIs	Lib Software User Manual	4.5
C, C++, Java APIs	Orbit Software User Manual	4.5
C, C++, Java APIs	Pointing Software User Manual	4.5
C, C++, Java APIs	Visibility Software User Manual	4.5

The documentation is available for download and on-line browsing 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>

2.3 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 10.5.x
 - Software Requirements: gcc compiler version 4.2.x

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, MACIN64
- 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

2.4 Installation Packages

The CFI libraries are provided as zip packages:

API	Package Name	MD5 Checksum
C	EOCFI-4.5-CLIB-LINUX32_LEGACY.zip	26a1195b0d3ad2f13fc61af0455f9b72
C	EOCFI-4.5-CLIB-LINUX64.zip	b30910e586eb5ba49e21dd7c2c145708
C	EOCFI-4.5-CLIB-LINUX64_LEGACY.zip	c3e90908da4c45524860d4012ffb63bc
C	EOCFI-4.5-CLIB-MACIN64.zip	c458229868be48ed3c541ace6e878da3

C	EOCFI-4.5-CLIB-WINDOWS.zip	2cc4c67a0181722a9c652fd4f746c2e0
C++	EOCFI-4.5-CPPLIB-LINUX32_LEGACY.zip	0979ad325ec9ecf811c54d37eb4f8617
C++	EOCFI-4.5-CPPLIB-LINUX64.zip	b2597816e939f69aaf79dc319d2c87fa
C++	EOCFI-4.5-CPPLIB-LINUX64_LEGACY.zip	6766b06e2dfb2d5a1c7cd4510eb01e2c
C++	EOCFI-4.5-CPPLIB-MACIN64.zip	cae6ef073d54ce3aef9fe0cd6d49eea7
C++ (*)	EOCFI-4.5-CPPLIB-WINDOWS_DLL.zip	54247b98bcfc7227994241452f7d79d5
C++ (**)	EOCFI-4.5-CPPLIB-WINDOWS_STA.zip	7cdb6700b497cc9316c724610232b725
JAVA	EOCFI-4.5-JAVALIB-LINUX32_LEGACY.zip	757cd8f117442a59bf190d056da5e08a
JAVA	EOCFI-4.5-JAVALIB-LINUX64.zip	860871b691c86b18e70de76cb095b974
JAVA	EOCFI-4.5-JAVALIB-LINUX64_LEGACY.zip	347be65c380bb0f6151d37a91c062a7a
JAVA	EOCFI-4.5-JAVALIB-MACIN64.zip	bcca60fb24ada1fe0a9a7c99d776c8cc
JAVA	EOCFI-4.5-JAVALIB-WINDOWS.zip	c821a3047c1bba415c505523e2dbfdd3

(*) 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>

2.5 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 and Section 6 “LIBRARY USAGE” of each Library User Manual.

In order to be able to use the XML validation function in the explorer_data_handling library, it is necessary to install the xerces libraries and the SAX2Count binary. The PATH environment variable shall be pointing at the SAX2Count location.

As of version 4.3, dynamic linking to libxml2 external libraries is no longer required.

As of version 4.5, user applications using the pointing library need to be built with openmp support (adding -fopenmp switch in gcc).

3 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.

3.1 File Handling Library

- **Limitation to 512 characters in XML fields removed.**
Now the user can read or write strings of any length. For reading functions, as memory is allocated within the CFI function, it is the user responsibility to deallocate such memory.

3.2 Data Handling Library

- **New DEM configuration file.**
New tags in DEM configuration file to support DEM minitiles definition and geoid computation. The user can now:
 - Enable the extended DEM intersection algorithm;
 - Configure number of harmonics used in geoid computation for ACE2 DEM;
 - Disable geoid computation for ACE2 DEM.The DEM configuration file is described in section 9.12 of the Data Handling SUM.
- **EarthCARE support.**
Files are generated according to File Format Standard 2.0.

3.3 Lib Library

- **New function `xl_geoid_calc` to compute the height of the geoid over the ellipsoid.**
The geoid is computed at a given longitude and latitude according to the input model (default is EGM96 model). EGM96 is a geopotential model of the Earth consisting of spherical harmonic coefficients complete to degree and order 360. The user can define the number of harmonics to be used for the computation (if this number is less than 360, the result is less accurate but the computation time is shorter).
The equivalent method for C++ and JAVA APIs is **`GeoidCalcInputs::geoidCalc`**.

3.4 Orbit Library

- **New function for filtering Orbit Files.**
The function `xo_orbit_data_filter` provides the interface of a generic filter for orbit data (state vectors). The filter implemented in this release is the outlier filter that allows discarding state vectors exceeding of a given threshold the state vector expected by interpolation. The function is described in Section 7.58 of the Orbit User Manual.
The equivalent method for C++ and JAVA APIs is **`OrbitFunc::orbitDataFilter`**.
- **Improvement in `xo_osv_compute`.**
Acceleration is now computed also with TLE and precise propagation.

3.5 Pointing Library

- **New Target list extra functions.**
A set of new functions is available to compute target extra results from a list of targets previously computed using function `xp_target_list_inter` (for example, `xp_target_list_extra_vector`, see

Pointing User Manual, Section 7.87). These extra functions can be used in order to improve runtime performance when a large number of computations is requested. The computation is optimized with respect to the single extra function and parallelized via multithreading (openmp). See also note on multithreading at Section 7.87.1.1.

The equivalent methods for C++ and JAVA APIs have been added in the Target class (e.g. **Target::extraListVector**).

- **DEM intersection algorithm extended.**

The DEM intersection algorithm has been extended. The extension is based on the fact that wide areas of the dataset can be skipped if the Line of Sight is higher than the maximum altitude of the DEM itself. The user can now compute offline such maximum altitudes and store them in a file using function **xp_gen_dem_max_altitude_file** (see Section 7.102 of Pointing User Manual). The extension can be enabled via the DEM configuration file (see above Section 3.2).

PointingFunc::genMaxAltFile is the equivalent method for C++ and JAVA APIs.

- **Runtime performance improvement with ACE2 DEM datasets.**

Altitudes in ACE2 datasets are expressed w.r.t. EGM96 geoid. In order to compute altitudes w.r.t. the ellipsoid, the CFI Software needs to compute at run-time, at the corresponding longitude and latitude, the geoid height w.r.t. the ellipsoid. This can be expensive for runtime especially when a large number of geolocations is computed. The user has two options:

- Reduce the number of harmonics used to compute the geoid: this can be achieved setting the number of harmonics in the DEM configuration file. The result is less accurate but the runtime performance is improved;
- Generate (offline) a new dataset based on the original ACE2 dataset but with altitudes expressed w.r.t. the ellipsoid. This can be achieved using the new function **xp_gen_dem_altitudes_from_ellipsoid** (see Pointing User Manual, section 7.103). The user needs to disable the geoid computation in the DEM configuration file. In this way the geoid is not computed at runtime.

PointingFunc::genDemAltitudesFromEllipsoid is the equivalent method for C++ and JAVA APIs.

4 SOLVED PROBLEMS

The following Anomalies have been solved:

ANR Id	Description
515	XX_Init_Err: mutex contention slows down multithread applications. <i>XX_Init_Err internal function is called several times to verify that the error table has been initialized or not. This is not efficient especially because a mutex is locked/unlocked to protect a flag indicating if the error tables have been initialized. When a multithreaded app runs, this mutex is contended very often and the threads are blocked. In order to fix this issue, error tables are now initialized statically.</i>
514	If number of harmonics is set to zero in an OSF, the file does not validate against latest version of schema.
513	xo_orbit_init_file: error with Restituted Orbit File and first OSV close to previous ANX (Issue

	reported by Sentinel-1 project).
512	xo_orbit_init_file: error with Restituted Orbit File and last OSV close to next ANX (Issue reported by Sentinel-1 project).
510	xv_station_vis_time: wrong computation when search interval > cycle length (Issue Reported by Sentinel-1 project).
509	Mismatch between Reference Frames identifiers in different libraries / APIs.
507	Memory leak in xo_osv_compute (Issue reported by EarthCARE project). <i>Remark: the problem was due to a wrong usage of the input model_id when not initialized. The user has now to explicitly initialized the model_id with xl_model_init function. If the model_id is passed uninitialized to functions, the function will use a temporary model id set to default model.</i>
501	gen_osf_append_orbit_change: unexpected error returned when the MLST jump exceeds the maximum allowed difference. <i>Remark: a warning is now issued in case of MLST and longitude discontinuities exceeding the acceptable thresholds.</i>
495	Computation of UT in precise propagator is not correct.
491	Orbit Propagator fails when orbit initialize with Doris file plus Orbit File (Issue reported by the SMAT development team).
490	xo_time_to_orbit: error when orbit_id initialized with single OSV.
485	Data Handling writing functions: Creator Version defined in input fixed header is overwritten by the CFI. <i>Remark: now the user can pass to the function his own value for the Version. If the input is left empty, the CFI version is written instead.</i>
481	Elevation angle computation not correct if target computed using quaternions file.
480	False gap detected when reading DORIS Navigator files.