EARTH OBSERVATION MISSION CFI SOFTWARE

Release Notes –Version 4.9

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

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	4.9	30 April 2015
Data Handling	4.9	30 April 2015
Lib	4.9	30 April 2015
Orbit	4.9	30 April 2015
Pointing	4.9	30 April 2015
Visibility	4.9	30 April 2015
EECommon (*)	4.9	30 April 2015

(*) 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	4.9
General	General Software User Manual	4.9
C API	Quick Start Guide	4.9
C API	File Handling Software User Manual	4.9
C API	Data Handling Software User Manual	4.9
C API	Lib Software User Manual	4.9
C API	Orbit Software User Manual	4.9
C API	Pointing Software User Manual	4.9
C API	Visibility Software User Manual	4.9

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 2010 Express or Professional edition)

- **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 anf 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 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 6 for all platforms

3.4 Installation Packages

The CFI libraries are provided as zip packages:

API	Package Name	MD5 Checksum
C	EOCFI-4.9-CLIB-LINUX32_LEGACY.zip	16137da4e19e03628552b90600450b12
C	EOCFI-4.9-CLIB-LINUX64.zip	f4dd956decb82adb7992591b4d2e6256
C	EOCFI-4.9-CLIB-LINUX64_LEGACY.zip	b11947c9e1572f8fe3dbc776536e9384
C	EOCFI-4.9-CLIB-MACIN64.zip	be25eab3150b2dc190bcadd9b384c4fe
C	EOCFI-4.9-CLIB-WINDOWS.zip	07a456305c827d0ac2b22fb930f16614
C++	EOCFI-4.9-CPPLIB-LINUX32_LEGACY.zip	34adbf52597505434ec73e3d5c606952
C++	EOCFI-4.9-CPPLIB-LINUX64.zip	d6b88a8f8ff61a7fd9f13a47dbe0a9e4
C++	EOCFI-4.9-CPPLIB-LINUX64_LEGACY.zip	62ba66f5bcfbadf6303355e3e93b090b
C++	EOCFI-4.9-CPPLIB-MACIN64.zip	c1416fc29f7bf17869b836640cf31f2c
C++	EOCFI-4.9-CPPLIB-WINDOWS_DLL.zip (*)	4d58c78aa1ea988fb5afc23c366ddf51
C++	EOCFI-4.9-CPPLIB-WINDOWS_STA.zip (**)	1e2c476bbcd5ddebff833c4dcaf48040
JAVA	EOCFI-4.9-DOC-CLIB.zip	1cf9a940123de003aa6df23d7e614875
JAVA	EOCFI-4.9-DOC-CPPLIB.zip	a9782a95ec089f698322e975a37a2a5c
JAVA	EOCFI-4.9-DOC-JAVALIB.zip	665ca17d49cfcec914bd7758399faff4
JAVA	EOCFI-4.9-JAVALIB-LINUX32_LEGACY.zip	75b909aabd1d5f8cbcea43cc396039cf
JAVA	EOCFI-4.9-JAVALIB-LINUX64.zip	fd067fd38f7af96b20aa6c0a8a0a0978

(*) 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 no longer required by the XML validation function and tool in the Data Handling library.

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 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 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.

In WINDOWS platforms, OpenMP is not supported in Visual Studio 2010 Express, therefore no additional switch is required. Functions will operate in single-threading mode.

The XML validation function and tool in the Data Handling library use the libxml2 library. For WINDOWS platform, 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.

- **Dependency to third-party libraries**
  As mentioned in section 3, the EOCFI depends on 3rd party libraries that are provided within the distribution package as separated libraries. The linker needs to be instructed to link against such libraries, e.g. for Linux C API the following additional flags need to be used:
  ```
  -lgeotiff -ltiff -lproj -lxml2 -lm -lc -lpthread
  ```
  Detailed instructions for building an application using the EOCFI libraries are provided in Section 6 of each Library User Manual.

- **Support for CCSDS Orbit Ephemeris Message (OEM) files**
  - **Data Handling Library:** new function to read OEM files: **xd_read_oem**. The format of OEM files is described in http://public.ccsds.org/publications/archive/502x0b2c1.pdf .

Section 9.21 of the Data Handling Software User Manual describes which items of the OEM file are read by xd_read_oem.

- **Orbit Library:** functions **xo_orbit_init_file** and **xo_orbit_id_init** have been extended to initialize an orbit_id using OEM files.

- **Improvement in dem_id initialization**
    - **Pointing Library:** improvements in **xp_dem_init**
        - The DEM files are looked for in the directory specified in the field Directory in the DEM configuration file. If this field is empty, the DEM files are looked for in the directory where the DEM configuration file is placed.
        - The input Digital Elevation Model initialization mode is ignored and the one in the DEM configuration file is used instead.

5 SOLVED PROBLEMS

The following Anomalies have been solved:

ANR Id	Description
558	Java SatId object does not work when satId initialized with satcfg file
572	Problems with IERS bulletin A file format
573	xv_zonevistime_compute: problem with swath_id and attitude_def inputs *(this issue has been reported by the Sentinel-2 Mission Planning Team)*
586	xv_time_segments_mapping_compute: incomplete mappings *(this issue has been reported by the Sentinel-2 Mission Planning Team)*
587	xv_time_segments_mapping_compute: incorrect coverage computation *(this issue has been reported by the Sentinel-2 Mission Planning Team)*
588	Errors using xv_zonevistime_coverage *(this issue has been reported by the Sentinel-2 Mission Planning Team)*
600	xp_attitude_transform does not work properly with Star Tracker Files
605	zone visibility computation: zone vertices are sporadically ignored *(this issue has been reported by the Sentinel-2 Mission Planning Team)*
606	Memory Leak in Java/C++ Attitude/Target objects *(this issue has been reported by the EarthCARE Project)*

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