EARTH OBSERVATION MISSION CFI SOFTWARE

Release Notes –Version 4.11

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	4.11	18 April 2016
Data Handling	4.11	18 April 2016
Lib	4.11	18 April 2016
Orbit	4.11	18 April 2016
Pointing	4.11	18 April 2016
Visibility	4.11	18 April 2016
EECommon (*)	4.11	18 April 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	4.11
General	General Software User Manual	4.11
C API	Quick Start Guide	4.11
C API	File Handling Software User Manual	4.11
C API	Data Handling Software User Manual	4.11
C API	Lib Software User Manual	4.11
C API	Orbit Software User Manual	4.11
C API	Pointing Software User Manual	4.11
C API	Visibility Software User Manual	4.11

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)

- **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-4.11-CLIB-LINUX32_LEGACY.zip	931c19fbd50b162a36338cb95c6d5570
C	EOCFI-4.11-CLIB-LINUX64.zip	97b44a25591d2899b3750ffa29114fdd
C	EOCFI-4.11-CLIB-LINUX64_LEGACY.zip	4d8e08c8d8cc7ebbbf9a1fbe7cc0fbc3
C	EOCFI-4.11-CLIB-MACIN64.zip	2f3893f3b1932ac60d9ab3e942b5d126
C	EOCFI-4.11-CLIB-WINDOWS32.zip	57c4c0895083512d7f4ac054be798f86
C	EOCFI-4.11-CLIB-WINDOWS64.zip	304d44cd64a9140b9b515bc28badd00d
C++	EOCFI-4.11-CPPLIB-LINUX32_LEGACY.zip	d3438a5e721968b6720512e1783638df
C++	EOCFI-4.11-CPPLIB-LINUX64.zip	32b1299ce1702a88b575b8e35c1e9720
C++	EOCFI-4.11-CPPLIB-LINUX64_LEGACY.zip	ac7a84a18743fe2dcd91b6039b4e939b
C++	EOCFI-4.11-CPPLIB-MACIN64.zip	ee0ec35c2d6442c537bc1142491cdc8c
C++	EOCFI-4.11-CPPLIB-WINDOWS32_DLL.zip (*)	61030027f2135dfc08f87f846a3cfb18
C++	EOCFI-4.11-CPPLIB-WINDOWS32_STA.zip (**)	2c494db40bf88adafa72fbd0e8e79753
C++	EOCFI-4.11-CPPLIB-WINDOWS64_DLL.zip (*)	261e48d162ea3b52529c0f368f1e6613
C++	EOCFI-4.11-CPPLIB-WINDOWS64_STA.zip (**)	cbdc7ccd55b45cf999f7fca536c30bb3
JAVA	EOCFI-4.11-JAVALIB-LINUX32_LEGACY.zip	99673375d0399fcdab6a8fcc08b86e9f
JAVA	EOCFI-4.11-JAVALIB-LINUX64.zip	c4d2ec3df1a28eb782cdd62a8a20a7bb
JAVA	EOCFI-4.11-JAVALIB-LINUX64_LEGACY.zip	b5fcb8d4d64739c98e0841de1bfddb16
JAVA	EOCFI-4.11-JAVALIB-MACIN64.zip	e9c8b21f919db5ca15eb60c2dc06d449
JAVA	EOCFI-4.11-JAVALIB-WINDOWS32.zip	bae87994769c823e3b06078a4c91e8e1
JAVA	EOCFI-4.11-JAVALIB-WINDOWS64.zip	ffeb77a3a70aa6ca913dd8d67e3e88c4

(*) 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 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.

- **New platform: Windows 64-bit.**
  It is now possible to use the EOCFI SW to build applications in Windows 64-bit platforms.

- **Java API: upgrade to Java version 8.**
  Java 8 SDK / JRE are now required to build and run applications using the JAVA API.

- **Upgrade to libxml2 version 2.9.2.**
  The EOCFI SW uses now a more recent version of libxml2.

- **New Missions supported: BIOMASS, SAOCOM-CS, Sentinel-5.**
  Mission identifiers for the new supported missions are available.
  (this feature has been requested by BIOMASS, SAOCOM-CS and Sentinel-5 projects respectively).

- **Aeolus Mission: orbit parameters changed according to new orbit requirements.**
  (this feature has been requested by The Aeolus project).

- **TLE designators updated for Sentinel-1A/2A/3A.**

- **New function to extract attitude data (quaternions or roll-pitch-yaw angles) from attitude_id.**
  - **Pointing Library:** the function `xp_get_attitude_data` receives as input an attitude_id and extracts the corresponding attitude data (time and quaternions/roll-pitch-yaw angles). See section 7.54 of the Pointing SUM.

- **New function to generate attitude data (list of quaternions and roll-pitch-yaw angles).**
  - **Pointing Library:** the function `xp_gen_attitude_data` generates a list of attitude records (i.e. time and quaternions/roll-pitch-yaw angles). Inputs for the function are an orbit_id and an attitude definition structure. The command line tool `gen_attitude` is provided within the distribution package, it generates attitude files given as input an orbit file and an attitude definition file. See section 7.55 of the Pointing SUM.

- **Support for new DEM type: ACE2 3 arcsec.**
  - **Data Handling Library:** New DEM Configuration File format (the new value ACE2_3SEC is allowed within the `<Dataset_Model>` tag).
  - **Pointing Library:** Pointing functions have been adapted to support ACE2 3 arcsec DEM (see http://tethys.eaprs.cse.dmu.ac.uk/ACE2/shared/overview)

5 SOLVED PROBLEMS

The following Anomalies have been solved:

ANR Id	Description
610	xo_orbit_id_init: crash in ANX computation (XL_Car_Kep: semi-major axis undefined).
630	xo_orbit_init_def: date is not set correctly.
631	GEO satellites: OsvComputeExtra gives wrong results.
633	xp_dem_init: wrong message string.

	(this problem has been reported by the S3 IPF Team).
645	xv_zonevistime_compute: wrong start/stop time with swath/zone covering pole.
646	Error in DEM intersection algorithm. *(this problem has been reported by the S3 IPF Team).* *NOTE: the specific reported problem has been identified and corrected. The intersection algorithm robustness has been improved so that, in the unlikely event that the DEM intersection algorithm fails, an approximated result is provided and a warning is returned.*

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