EARTH OBSERVATION MISSION CFI SOFTWARE

Release Notes –Version 4.4

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	4.4	5 July 2012
Data Handling	4.4	5 July 2012
Lib	4.4	5 July 2012
Orbit	4.4	5 July 2012
Pointing	4.4	5 July 2012
Visibility	4.4	5 July 2012
EECommon (*)	4.4	5 July 2012

(*) 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	4.4
General	General Software User Manual	4.4
C API	Quick Start Guide	4.4
C API	File Handling Software User Manual	4.4
C API	Data Handling Software User Manual	4.4
C API	Lib Software User Manual	4.4
C API	Orbit Software User Manual	4.4
C API	Pointing Software User Manual	4.4
C API	Visibility Software User Manual	4.4

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)

- **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-4.4-CLIB-LINUX32_LEGACY.zip	b3dbc340265f04fddb61491b1cff0f24
C	EOCFI-4.4-CLIB-LINUX64.zip	70c8de6c1a89215ae1f22bff7109def5
C	EOCFI-4.4-CLIB-LINUX64_LEGACY.zip	e32426c168dad3116ec4657c9e974b78
C	EOCFI-4.4-CLIB-MACIN64.zip	c9ede232375fbb706bd9aec9e4aac04d

C	EOCFI-4.4-CLIB-WINDOWS.zip	f3daf7a60c2f190e2807ca9728ff04bb
C++	EOCFI-4.4-CPPLIB-LINUX32_LEGACY.zip	80ec35f9327e3581632a99c226001b6d
C++	EOCFI-4.4-CPPLIB-LINUX64.zip	4e86e7ddc587933184a99494980efd80
C++	EOCFI-4.4-CPPLIB-LINUX64_LEGACY.zip	393e1a91043d44f8b977804a2e552932
C++	EOCFI-4.4-CPPLIB-MACIN64.zip	31dd4320f5f2ac4fc0965149280fe9ee
C++	EOCFI-4.4-CPPLIB-WINDOWS_DLL.zip (*)	cdf7ff6fdff447a88e2030a8a323ee84
C++	EOCFI-4.4-CPPLIB-WINDOWS_STA.zip (**)	b43bcf6cdd2c8fb42fffe0cff64f627e
JAVA	EOCFI-4.4-JAVALIB-LINUX32_LEGACY.zip	99fe1781ffd634ce79f788bf63076e70
JAVA	EOCFI-4.4-JAVALIB-LINUX64.zip	9be1a9a5c8ed68084e9311e54b527204
JAVA	EOCFI-4.4-JAVALIB-LINUX64_LEGACY.zip	379dbe716554b6b70a51bd3b6f1f1f67
JAVA	EOCFI-4.4-JAVALIB-MACIN64.zip	e2fa43dcf9a454b22569123db7f4e283
JAVA	EOCFI-4.4-JAVALIB-WINDOWS.zip	08531a21feed269555bebf4c6a5d5676

(*) 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.

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 Data Handling Library

1) **New DEM configuration file: New tags in DEM configuration file to support DEM memory cache.**
   The user can now select and configure different methods to access the DEM dataset (direct I/O

access to file or memory caching).
The DEM configuration file is described in section 9.12 of the Data Handling SUM.
Guidelines to select and configure the access methods are in section 7.61.2 of the Pointing SUM.

3.2 Lib Library

1) **New non-iterative method to compute transformation from Cartesian to Geodetic coordinates.**
Function **xl_cart_to_geod** has been extended. It is possible to compute conversions with two different methods, one iterative already implemented in previous versions and one (new) direct.
More details can be found in section 7.25 of the Lib SUM.
The correspondent method in the C++ and JAVA APIs is **EECFI::Coord::getGeodetic**.

2) **New Reference Frames for position on orbit computation.**
Function **xl_position_on_orbit** has been extended (see section 7.41 of Lib SUM). It is possible to compute position on orbit with respect to:

    a. Earth Fixed Reference Frame (OPS angle).

    b. Geocentric Mean of 2000 Reference Frame.

    The correspondent method in the C++ and JAVA APIs is **EECFI::StateVector::getPositionOnOrbit**.

3.3 Orbit Library

1) **Support for geostationary satellites.**
The following new functions are available:

    a. **xo_orbit_init_geo**: orbit initialization (see section 7.8 of Orbit SUM).

    b. **xo_get_geo_orbit_info / xo_set_geo_orbit_info**: functions to get/set orbit data (see section 7.27 / 7.28 of Orbit SUM).

    The correspondent methods in the C++ and JAVA APIs are **EECFI::OrbitId::init, EECFI::OrbitId::getGeoOrbitInfo, EECFI::OrbitId::setGeoOrbitInfo**

2) **New function to compute the time, position and velocity vectors in Earth-Fixed associated to a given position on orbit.**
The new function **xo_position_on_orbit_to_time** is available (see section 7.57 of Orbit SUM).
The correspondent method in the C++ and JAVA APIs is **EECFI::OrbitId::positionOnOrbitToTime**.

3.4 Pointing Library

1) **New attitude mode for geostationary satellites: Yaw flip mode.**
The new nominal attitude model **XP_MODEL_GEO** is now available. See sections 7.2 and 7.2.3.1 of Pointing SUM for further details.

2) **New function to compute pointing data between two satellites.**
The new function **xp_target_sc** is available (see section 7.82 of Pointing SUM).
The correspondent method in the C++ and JAVA APIs is **EECFI::Target::targetSc**.

3) **New target function to compute intersections of a set of lines of sight (list, strip, grid).**
The new function **xp_target_list_inter** is available (see section 7.85 of Pointing SUM).
The correspondent method in the C++ and JAVA APIs is **EECFI::Target::targetListInter**.

4) **Runtime performance improvements on computation of intersections with the DEM**

   a. Conversions from Cartesian to Geodetic coordinates are using now the direct method described in section 7.25 of the Lib SUM. This method is much faster than the iterative one.

   b. Computation of geodetic distances has been changed in order to improve runtime performances.

   c. Proper configuration of the DEM access method (see section 3.1) can improve runtime performances. See section 7.61.2 of the Pointing SUM for more details.

3.5 Visibility Library

1) **New visibility function for computations of visibility segments from one satellite to another**.

   The new function **xv_sc_vis_time** is available (see section 7.5 of Visibility SUM).
   The correspondent method in the C++ and JAVA APIs is **EECFI::Swath::scVisTime**.

4 KNOWN LIMITATIONS

The function **xp_target_list_inter** has the following limitations that will be removed in future releases:

1) The maximum number of lines of sight is 30.
2) In the grid mode: the grid number of points shall be the same on both dimensions.