



Code: EOCFI-DMS-SRN-000  
 Date: 31/01/2011  
 Issue: 4.2

## Main Page

[Introduction](#)

[Release Description](#)

[Software Requirements](#)

[Installation](#)

[Software Version](#)

# Earth Observation Mission CFI Software JAVA Libraries. Release Notes - Version 4.2

## INTRODUCTION

This note describes the changes introduced in the current release of the Earth Observation CFI software JAVA libraries. This note consists of the following sections:

- [Release Description](#)
- [Software Requirements](#)
- [Installation](#)
  - [CFI packages and installation](#)
  - [How to build the validation](#)
  - [How to build the example](#)
- [Software Version](#)
  - [New Functionalities](#)
  - [Closed SPR's](#)
  - [Known Problems](#)

## RELEASE DESCRIPTION

The current release of the Earth Observation CFI consist of the following items:

- CFI JAVA libraries: The libraries consist on a library wrapper of the Earth Observation CFI C libraries (version 4.2) so that for every C library, a JAVA Native library and a JAVA archive file (JAR files) is distributed:

Library	Issue	Date
ECommon	4.2	31/01/11
FileHandling	4.2	31/01/11
DataHandling	4.2	31/01/11
Lib	4.2	31/01/11
Orbit	4.2	31/01/11
Pointing	4.2	31/01/11
Visibility	4.2	31/01/11

The library names depend on the OS (XXX = Library name):

OS	Java Native library	JAR file
LINUX and SOLARIS	libCfi.XXX.so	XXX.jar
MAC OS X	libCfi.XXX.jnilib	XXX.jar
WINDOWS	libJCfi.XXX.dll	XXX.jar

- User Manuals:

Title	Reference	Issue	Date
-------	-----------	-------	------

General Software User Manual	EO-MA-DMS-GS-019	4.2	31/01/11
EE Common Software User Manual	EO-MA-DMS-GS-020	4.2	31/01/11
FileHandling Software User Manual	EO-MA-DMS-GS-021	4.2	31/01/11
DataHandling Software User Manual	EO-MA-DMS-GS-022	4.2	31/01/11
Lib Software User Manual	EO-MA-DMS-GS-023	4.2	31/01/11
Orbit Software User Manual	EO-MA-DMS-GS-024	4.2	31/01/11
Pointing Software User Manual	EO-MA-DMS-GS-025	4.2	31/01/11
Visibility Software User Manual	EO-MA-DMS-GS-026	4.2	31/01/11
Software Usage Guide	EO-MA-DMS-GS-027	4.2	31/01/11

The libraries have been created with the following computer systems:

Machine	OS
Sun under Solaris 64-Bits	Solaris 5.9
PC under Linux 32-Bits	Linux 2.6.16 (RedHat)
PC under Linux 64-Bits	Linux 2.6.16 (RedHat)
Macintosh under MacOS X 32-Bits	Mac OS X 10.4.6
Macintosh with Intel processor under MacOS X 32-Bits	Mac OS X 10.4.9
Macintosh with Intel processor under MacOS X 64-Bits	Mac OS X 10.5.7
PC under Windows XP	Microsoft Windows XP

## SOFTWARE REQUIREMENTS

The following table shows for every OS the list of additional libraries that are needed in order to link an application with the Earth Observation CFI libraries.

OS	Compiler	External libraries
Solaris 64-Bits	Java SE 5	libxml2 version 2.4.23 or later
		libpthread.so (POSIX thread library)
Linux 32-Bits	Java SE 5	libxml2 version 2.6.23 or later
		libpthread.so: POSIX thread library
		glibc 2.4
		GLIBCXX 3.4.9
Linux 64-Bits	Java SE 5	libxml2 version 2.6.23 or later
		libpthread.so: POSIX thread library
		glibc 2.4
		GLIBCXX 3.4.9
Mac OS X 32-Bits	Java 1.5.0_26	libxml2 version 2.6.22.
		libpthread.so: POSIX thread library
Mac OS X Intel 32-Bits	Java 1.5.0_19	libxml2 version 2.6.16
		libpthread.so: POSIX thread library
Mac OS X Intel 64-Bits	Java 1.5.0_16	libxml2 version 2.6.22.
		libpthread.so: POSIX thread library
MS Windows(*)	Java SE 5	libxml2 version 2.6.20 or later (including iconv-1.9.1 and zlib-1.2.3)
		pthread.lib: POSIX thread library

(\*) When linking an application for Windows using the CFI dynamic libraries, the linking with the LIBC.LIB or LIBCM.LIB should be avoided. The library MSVCRT.DLL should be used instead. This library is not provided with the CFI installation package, as it is included with the Windows OS (It also can be downloaded from: <http://support.microsoft.com/default.aspx?scid=kb;en-us;259403>).

## CFI INSTALLATION

## CFI Packages and Installation

The CFI software and documents can be downloaded from the ESA EOP System Support Division Web Server:

<http://eop-cfi.esa.int> (main page)

From there, just follow the links until you reach the Earth Observation CFI page:

[http://eop-cfi.esa.int/CFI/ee\\_cfi\\_software.html](http://eop-cfi.esa.int/CFI/ee_cfi_software.html) Follow the instructions given on the page and you will be able to save the distribution file(s) on your local disk.

The CFI libraries are provided in different formats depending on the platform:

- WINDOWS and MAC OS: Installation programs. This program will guide you through the installation process.  
These programs will install the documents together with software.
- Other platforms: EOCFI\_JAVA\_version\_OS.tar.gz (OS = LINUX, LINUX64, SOLARIS64 The package has to be uncompressed manually:
  - > gunzip EOCFI\_JAVA\_version\_OS.tar.gz
  - > tar xvf EOCFI\_JAVA\_version\_OS.tar

Upon completion the installation procedure, the following items will be on your computer:

- Libraries
- Include files
- Example programs
- Validation programs
- Software User manuals (only for WINDOWS and MAC OS, for other OS they can be downloaded from the [ESA EOP Web Server](#).)
- cfi\_tools: auxiliary libraries required for the CFI (only for WINDOWS).

Finally, the validation program should be compiled and run to check that the installation has been successful.

### How to build the validation

For every library the CFI installation contains a directory called "validation". This directory contains the validation program (*LibraryNameValid.java*) and associated validation script (*validate.bat* for WINDOWS and *validate.sh* for the rest of the systems). This program should be run to verify the proper installation of the CFI Library. The procedure is the following:

1. Go to directory validation
2. Edit the validation script and configure it to your installation if necessary.
3. Note in particular that if the CFI requires to link with other CFIs, you will have to specify the location of those other CFI libraries. If, when installing those other CFIs, you always followed the advice given below, this will be easier.
4. Run the validation program using:
  - For all OS except Windows: `./validate.sh OS` where OS stands for the different allowed operative systems.
  - For Windows: `validate.bat`

The validation program is created, executed and a validation status message printed. The message should look like:

```
LibraryName: ... CFI LIBRARY INSTALLATION = OK
```

or:

```
LibraryName: ... CFI LIBRARY INSTALLATION = FAILED !!!
```

In the latter case, check again your installation, and run the validation program again if necessary. If the message persists, report the problem.

During the execution of the validation program a log file *LibraryNameValid.OS.out* (OS stands for the different allowed operative systems) is also created. It can be consulted for a detailed listing of the validation run.

### How to build the example

For every library the CFI installation contains a directory called "example".

This directory contains example program (*LibraryNameExample.cpp*), the associated compilation script (*runExample.bat* for WINDOWS and *runExample.sh* for the rest of the operating systems) and the data files (under the data directory). The example program is provided to illustrate how the interface with the CFI functions works, and in particular how to handle the returned errors through exceptions.

The examples should be self-explanatory. To use them, use the same procedure as for the validation program. In an user application, the same conventions to compile and link as in the example compilation script should be

followed.

Note that native libraries are dynamic and, when using dynamic linking libraries, proper setting of the environment must be performed at run-time. This means:

- SOLARIS/Linux: adding to the LD\_LIBRARY\_PATH environment variable the locations of all dynamic libraries needed.
- MacOS: adding to the DYLD\_LIBRARY\_PATH environment variable the locations of all dynamic libraries needed.
- Windows XP: adding to the PATH environment variable the locations of all dynamic libraries needed.

## SOFTWARE VERSION

### New Functionalities

The following new features/requirements have been implemented:

General:

- NORAD TLE Satellite number designators for GOCE, SMOS and CRYOSAT2.
- New satellite identifiers: SENTINEL\_1C, SENTINEL\_2A, SENTINEL\_2B, SENTINEL\_2C, SENTINEL\_3A, SENTINEL\_3B, SENTINEL\_3C.
- Check normalization of input vectors, matrices and quaternions.
- Orbit Id initialisation: the state vector/orbital changes time stamps from the orbit files are recomputed to be compatible with the input time correlations (TimeCorrelation).

DataHandling:

- New DEM configuration file.
- New OSF and STF format to support curved Mean Local Solar Time.

Orbit:

- Support for curved Mean Local Solar Time. New functions:
  - New overloaded constructor of OrbitId with RefOrbitInfo as input
  - OrbitFunc::genOsf with RefOrbitInfo as input
  - OrbitFunc::genOsfAppendOrbitChange with RefOrbitInfo as input
  - OrbitFunc::genOsfChangeRepeatCycle with RefOrbitInfo as input

Pointing:

- Support for new DEM model: ACE2 9" resolution.
- New method DemId::getInfo to extract DEM data source flag.
- Possibility of using regional DEM.

Visibility:

- Support in visibility functions for TLE files and precise propagation.

## CLOSED ANOMALIES

### Software problems

The following Software problems have been fixed:

ANR Nr.	Description
418	The CFI algorithm for SAR modes produces wrong results when pointing to the left of the boresight direction (i.e. when target azimuth = 270.0 deg). In particular the left and right points are incorrectly geo-located.
419	DEM target computation fails when using ray tracing model.
432	Wrong error message returned by TimeCorrelation file constructor when called with a time interval outside the validity time of the input orbit file.
434	Empty list of station visibility segments when using OSF.
435	Error in Attitude::changeFrame when attitude frame contains offsets.
436	OrbitId::osvComputeExtra does not compute the osculating elements when used in combination with the precise orbit propagator.
437	OEF Format is obsolete, however is still described in the SUMs (e.g. Data Handling)

440	Instrument offsets not taken into account in target computation functions.
441	Yaw Steering Attitude Mode: Earth Fixed velocity should be computed at the spacecraft instead of nadir.
442	Failure in thread safety: some low level function mutexes cannot prevent simultaneous access to Id lists.
443	OrbitId file constructor fails with ROF (ANX not found).
444	For some missions, files are generated without Mission ID in the name
446	Memory leak in Attitude::changeFrame
451	Nodal period and seconds since ANX are computed incorrectly when using TLE
458	Typo in nominal eccentricity for Sentinel1B

## New features

The following new features have been implemented:

ANR Nr.	Description
220	Check on normalization of input vectors, matrices and quaternions.
347	MLST non linear drift supported (Linear, Quadratic and Harmonic terms).
383	New DEM dataset (ACE2 9SECS resolution) supported.
384	DEM Data source extraction supported.
402	Visibility computations supported with TLE and precise propagators.
406	Partial (regional) DEM datasets supported.
414	Cryosat-2, GOCE, SMOS supported with TLE propagator.
416	OrbitId file constructor: Re-computation of OSV times in case of discrepancy with time correlations defined in TimeCorrelation.
459	Support for File Format Standard version 2.0.

## Known Problems

The following precautions shall be taken into account when using the CFI software libraries:

CFI library	Problem	Work around solution
-	-	-