Main Page

Introduction   Release Description   Software Requirements   Installation   Software Version

Earth Observation Mission CFI Software
C++ Libraries. Release Notes - Version 4.1

INTRODUCTION

This note describes the changes introduced in the current release of the Earth Observation CFI software C++ 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 dynamic libraries for C++ (For Windows there exists a set of static libraries as well):

Library	Issue	Date
EECommon	4.1	07/05/10
FileHandling	4.1	07/05/10
DataHandling	4.1	07/05/10

Lib	4.1	07/05/10	
Orbit	4.1	07/05/10	
Pointing	4.1	07/05/10	
Visibility	4.1	07/05/10	

- User Manuals:

Title	Reference	Issue	Date
General Software User Manual	EO-MA-DMS-GS-017	4.1	07/05/10
EE Common Software User Manual	EO-MA-DMS-GS-010	4.1	07/05/10
FileHandling Software User Manual	EO-MA-DMS-GS-011	4.1	07/05/10
DataHandling Software User Manual	EO-MA-DMS-GS-012	4.1	07/05/10
Lib Software User Manual	EO-MA-DMS-GS-013	4.1	07/05/10
Orbit Software User Manual	EO-MA-DMS-GS-014	4.1	07/05/10
Pointing Software User Manual	EO-MA-DMS-GS-015	4.1	07/05/10
Visibility Software User Manual	EO-MA-DMS-GS-016	4.1	07/05/10

The libraries have been created with the following computer systems and compilers:

Machine	OS
Sun under Solaris 32-Bits	Solaris 5.7
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 under MacOS X 64-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 10.4.9
PC under Windows 95/98/NT/2000	Microsoft Windows 2000

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 32-Bits	g++ 4.2.2	libxml2 version 2.6.22 or later
		libpthread.so (POSIX thread library)

Solaris 64-Bits	g++ 4.2.2	libxml2 version 2.4.23 or later
		libpthread.so (POSIX thread library)
Linux 32-Bits	g++ 4.2.2	libxml2 version 2.6.23 or later
		libpthread.so: POSIX thread library
		glibc 2.4
		GLIBCXX 3.4.9
Linux 64-Bits	g++ 4.2.2	libxml2 version 2.6.23 or later
		libpthread.so: POSIX thread library
		glibc 2.4
		GLIBCXX 3.4.9
Mac OS X 32-Bits	g++ 4.2.1	libxml2 version 2.6.22.
		libpthread.so: POSIX thread library
Mac OS X 64-Bits	g++ 4.2.1	libxml2 version 2.6.22.
		libpthread.so: POSIX thread library
Mac OS X Intel 32-Bits	g++ 4.2.1	libxml2 version 2.6.16
		libpthread.so: POSIX thread library
Mac OS X Intel 64-Bits	g++ 4.2.1	libxml2 version 2.6.22.
		libpthread.so: POSIX thread library
MS Windows(*)	MS Visual C++ 8.0	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 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_C++_version_OS.tar.gz (OS = LINUX, LINUX64, SOLARIS, SOLARIS64 The package has to be uncompressed manually:

> gunzip EOCFI_C++_version_*OS*.tar.gz
> tar xvf EOCFI_C++_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 sucsessful.

How to build the validation

For every library the CFI installation contains a directory called "validation". This directory contains the validation program (*LibraryName*Valid.cpp) and associated makefile (make.*OS*). 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 makefile for your platform and configure it to your installation. The configuration parameters are all located at the top of the Makefile, with instructions on how to use them.
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 in section 6.3, this will be easier.
4. Run the validation program using:
   - For all OS except Windows: make -f make.OS where OS stands for the different allowed operative systems.
   - For Windows: nmake /f make.WINDOWS_DLL.mak (To link with the dynamic libraries)
     or
     nmake /f make.WINDOWS_STA.mak (To link with the static libraries)

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 *LibraryName*Valid.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 (*LibraryName*Example.cpp), the associated makefile (make.*OS*) 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 makefiles should be followed.
Note that CFI 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 95/98/NT/2000: adding to the PATH environment variable the locations of all dynamic libraries needed.

SOFTWARE VERSION

New Funcionalities

The following new features/requirements have been implemented:

DataHandling:

- GETASSEv2 DEM supported
- IERS B Bulletin format 2010 reading supported
- Ground Station DB 1.4 file reading supported

Lib:

- Time initialization with list of files
- Time initialization with OSF

Orbit:

- Time correlation compatibility check between time_id and orbit file data

Pointing:

- Pointing functions support DEM GETASSEv2
- Sentinel-1 attitude model (roll steering)
- Instrument offsets for attitude computations

Visibility:

- AOS/LOS mask mode from Ground Station DB 1.4 file
- Visibility across orbital changes

Closed SPR's

The following SPRs have been solved:

- EOCFI-SPR-137: Problem computing time_to_orbit when using a POF with SV that are just before the ANX and the input time is between the SV and the ANX.
- EOCFI-SPR-144: Typo EARHT in read and write attitude file.
- EOCFI-SPR-145: Interpolation problems with orbit files (102.XSD schema) whose OSV are not in EF.
- EOCFI-SPR-146: Target not found when using the DEM.
- EOCFI-SPR-147: Error in propagator initialization when using orbit_init_def.

Known Problems

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

CFI library	Problem	Work around solution
-	-	-

Generated on Wed Jul 21 2010 10:53:25 for by **doxygen** 1.7.1