BIBLOS USER MANUAL

BIBLOS

Prepared by: Karolina Szczepankiewicz

Approved by: Cristina de Negueruela

Authorized by: Lucia Soto

Code: GMV-EOM-UM

Internal Code: GMV 21126/16 V1/16

Version: 1.0

Date: 15/04/2016
## 1. CHANGE RECORD

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>05/04/2015</td>
<td>Initial version.</td>
</tr>
</tbody>
</table>

*Table 1-1: Change Record*
2. TABLE OF CONTENTS

1. CHANGE RECORD ..................................................................................................................... 2
2. TABLE OF CONTENTS .............................................................................................................. 3
3. LIST OF TABLES AND FIGURES ............................................................................................ 6
4. INTRODUCTION AND SCOPE ............................................................................................... 7
5. REFERENCES ............................................................................................................................. 8
   5.1. REFERENCE DOCUMENTS ................................................................................................. 8
   5.2. PROJECT DOCUMENTS .................................................................................................... 8
6. DESIGN ...................................................................................................................................... 9
   6.1. BLOCK ABSTRACT CLASS TEMPLATE ........................................................................... 9
       6.1.1. DESCRIPTION .............................................................................................................. 9
       6.1.2. TEMPLATE ARGUMENTS ......................................................................................... 9
       6.1.3. CONSTRUCTOR ARGUMENTS ................................................................................. 9
       6.1.4. FUNCTIONS TO IMPLEMENT ................................................................................. 9
   6.2. MODULE CLASS TEMPLATE ......................................................................................... 10
       6.2.1. PACKAGE ................................................................................................................. 10
       6.2.2. DESCRIPTION ......................................................................................................... 10
       6.2.3. TEMPLATE ARGUMENTS ....................................................................................... 10
       6.2.4. PUBLIC MEMBERS ............................................................................................... 10
7. CREATING BLOCKS AND MODULES ....................................................................................... 12
   7.1. CREATING NEW BLOCK .................................................................................................. 12
       7.1.1. GETTING ARGUMENTS FROM INPUT/OUTPUT VECTOR ....................................... 12
       7.1.2. PARSING CONFIGURATION FILES ......................................................................... 13
   7.2. CREATING NEW MODULE (CONNECTING BLOCKS) ....................................................... 13
8. INSTALLATION GUIDELINES .................................................................................................... 14
   8.1. SYSTEM REQUIREMENTS .............................................................................................. 14
       8.1.1. BLOCKS REQUIREMENTS ....................................................................................... 14
       8.1.2. MODULES REQUIREMENTS ............................................................................... 14
       8.1.3. BIBLOS.GEOMETRY REQUIREMENTS ................................................................. 14
       8.1.4. BIBLOS.SCENEGENERATOR REQUIREMENTS .................................................... 15
       8.1.5. BIBLOS.INSTRUMENT REQUIREMENTS ............................................................... 15
   8.2. FLAGS FOR BIBLOS INSTALLATION CMAKE ................................................................. 15
   8.3. BIBLOS.COTS .................................................................................................................... 16
       8.3.1. DESCRIPTION ............................................................................................................ 16
   8.4. TEST ................................................................................................................................... 16
9. BIBLOS BLOCKS AND MODULES ............................................................................................ 18
   9.1. ORBIT BLOCK .................................................................................................................... 18
       9.1.1. CONFIGURATON .................................................................................................... 18
       9.1.2. INPUT ....................................................................................................................... 18
       9.1.3. OUTPUT .................................................................................................................. 18
       9.1.4. EXAMPLE COMPILATION AND INSTALLATION COMMANDS ................................ 18
       9.1.5. EXAMPLE RUN COMMAND ................................................................................. 18
       9.1.6. EXAMPLE TEST COMMAND .............................................................................. 19
   9.2. ATTITUDE BLOCK .............................................................................................................. 19
       9.2.1. CONFIGURATON .................................................................................................... 19
       9.2.2. INPUT ....................................................................................................................... 19
9.9.5. EXAMPLE RUN COMMAND ................................................................. 28
9.9.6. EXAMPLE TEST COMMAND .............................................................. 29

9.10. SCENEGENERATOR MODULE ............................................................. 29
  9.10.1. CONFIGURATION ................................................................. 29
  9.10.2. INPUT ................................................................. 29
  9.10.3. OUTPUT ................................................................. 29
  9.10.4. EXAMPLE COMPILATION AND INSTALLATION COMMANDS .......... 29
  9.10.5. EXAMPLE RUN COMMAND .............................................................. 30
  9.10.6. EXAMPLE TEST COMMAND .............................................................. 30

9.11. INSTRUMENT MODULE ................................................................. 30
  9.11.1. CONFIGURATION ................................................................. 30
  9.11.2. INPUT ................................................................. 30
  9.11.3. OUTPUT ................................................................. 30
  9.11.4. EXAMPLE COMPILATION AND INSTALLATION COMMANDS .......... 30
  9.11.5. EXAMPLE RUN COMMAND .............................................................. 31
  9.11.6. EXAMPLE TEST COMMAND .............................................................. 31

10. STEP-BY-STEP INSTALLATION GUIDE .................................................... 32

  10.1. REQUIREMENTS FOR STEP-BY-STEP GUIDE ........................................... 32
  10.2. STEPS .......................................................................................... 32
    10.2.1. INSTALLATION OF COTS ......................................................... 32
    10.2.2. INSTALLATION OF BIBLOS LIBRARIES ........................................... 32
    10.2.3. INTEGRATION OF EXAMPLE SCENARIO AND POC WITH OPENSF, RUNNING EXAMPLE AND POC IN OPENSF 33
3. LIST OF TABLES AND FIGURES

TABLE 1-1: CHANGE RECORD ............................................................................................................. 2
TABLE 5-1: REFERENCE DOCUMENTS .................................................................................................. 8
TABLE 5-2: PROJECT DOCUMENTS ........................................................................................................ 8
TABLE 10-1: INSTALLATION OF COTS .................................................................................................. 32
TABLE 10-2: INSTALLATION OF ALL BIBLOS LIBRARIES ................................................................. 33
TABLE 10-3: INTEGRATION OF POC AND EXAMPLE SCENARIO DATABASE WITH OPENSF .......... 33
TABLE 10-4: RUNNING EXAMPLE SCENARIO IN OPENSF ............................................................... 34
TABLE 10-5: RUNNING POC IN OPENSF .............................................................................................. 34
4. INTRODUCTION AND SCOPE

This document provides the information needed to work with BIBLOS. A guide is provided to get started with BIBLOS, how to use it with existing software, or how to create new software units for BIBLOS.

In detail, the scope of this document is:

- In this document presented is the architecture of the Block (see 6.1) and Module (see 6.2) as well as helper classes for Module/Block parameter mapping (see 7.1.1) and parsing configuration files (see 7.1.2).
- Chapter 7. presents the quick tutorial on how to create a new block or a new module.
- For general installation guidelines and installation requirements please refer to chapter 8.
- Chapter 9. presents Biblos blocks and modules: the input and output arguments to the compiled binaries, installation, test and run example commands.
- For step-by-step installation guide for Biblos COTS and all Biblos libraries package, as well as openSF integration, please refer to chapter 10.
5. REFERENCES

5.1. REFERENCE DOCUMENTS

The following documents, although not part of this document, amplify or clarify its contents. Reference documents are those not applicable and referenced within this document. They are referenced in this document in the form [RD.X]:

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Title</th>
<th>Code</th>
<th>Version</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>[RD.1]</td>
<td>EO Missions and Elements Categorization</td>
<td>ARCHEO-E2E-TN-001</td>
<td>2.0</td>
<td>19/09/2012</td>
</tr>
</tbody>
</table>

Table 5-1: Reference Documents

5.2. PROJECT DOCUMENTS

The following documents are produced in the frame of this activity. They are referenced in this document in the form [PD.X]:

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Title</th>
<th>Code</th>
<th>Version</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>[PD.1]</td>
<td>EO models library roadmap</td>
<td>GMV-EOM-RM</td>
<td>3.0</td>
<td>25/02/2016</td>
</tr>
<tr>
<td>[PD.3]</td>
<td>EO models Design Definition File</td>
<td>GMV-EOM-DD</td>
<td>2.0</td>
<td>25/02/2016</td>
</tr>
<tr>
<td>[PD.4]</td>
<td>EO models Verification &amp; Validation Plan</td>
<td>GMV-EOM-VP</td>
<td>2.0</td>
<td>25/02/2016</td>
</tr>
<tr>
<td>[PD.6]</td>
<td>EO models Proof of Concept</td>
<td>GMV-EOM-UM</td>
<td>2.0</td>
<td>03/03/2016</td>
</tr>
</tbody>
</table>

Table 5-2: Project documents
6. DESIGN

This chapter describes the Biblos abstraction layer that the user should be aware about when designing or modifying blocks and/or modules.

6.1. BLOCK ABSTRACT CLASS TEMPLATE

Biblos.Block is an C++ abstract class template, which is a common base class for each block. First of all, Block provides the functions that are useful in implementing derived classes, such as helper functions and structure for mapping input parameters identifiers based on the order to their names. Block also declares the functions that need to be implemented by derived classes so the block could be executed in a module.

Block abstract class template referred to as Biblos.Block is available in Biblos.Commons.

6.1.1. DESCRIPTION

This is a common base class for Biblos blocks. All blocks that user want to execute within a module should inherit from Biblos.Block.

6.1.2. TEMPLATE ARGUMENTS

Declaration of Biblos.Block:

```cpp
template<typename GlobalConfigType, typename LocalConfigType> class Block;
```

Template arguments are:

- Type of global configuration file
- Type of local configuration file

Global configuration file structure should implement:

**Function:** static void read(const std::string &filename, B_GlobalConfigType & config, const std::string &line);

This function parses configuration file and fills config structure

**Function:** void log()const;

This function uses Logger from EOCFI Libraries to log configuration parameters

Local configuration file structure should implement:

**Function:** static void read(const std::string &filename, B_LocalConfigType & config, const std::string &line);

This function parses configuration file and fills config structure

**Function:** void log()const;

This function uses Logger from EOCFI Libraries to log configuration parameters

6.1.3. CONSTRUCTOR ARGUMENTS

Block(ParameterIOFileIds && ios)

A pair of vectors. First of them is a vector of input file identifiers. Second is a vector of output file identifiers. Identifier type is: std::string.

6.1.4. FUNCTIONS TO IMPLEMENT

**Function:** B_execute

```cpp
virtual int B_execute(const GlobalConfigType &global_param, const LocalConfigType &local_param,
                      const std::vector<std::string> & input, const std::vector<std::string> & output) = 0;
```

This is a function that executes block.

Parameters are:

- global_param – global configuration file
- local_param – local configuration file
input – a vector of block input file names
output – a vector of block output file names
The function process one detector line.

**Function:**

```
virtual std::map<DetectorLineID, std::pair<GlobalConfigType, LocalConfigType>>
parseConfigurationFiles (const std::string &globalConfigurationFilename, const std::string &
localConfigurationFilename) const = 0;
```

This is a function that parses configuration files.

Arguments are:

- `globalConfigurationFilename` – global configuration file name
- `localConfigurationFilename` – local configuration file name

Return value:

Map with the mapping of particular detector line identifier to its configuration – the global and the local one.

### 6.2. MODULE CLASS TEMPLATE

#### 6.2.1. PACKAGE

Module abstract class template referred to as Biblos. Module is available in Biblos.Commons package.

#### 6.2.2. DESCRIPTION

This is a common base class for Biblos modules. Module variadic template (a template with take a variable number of arguments) inherits from all the blocks that specialize that template. Object of a specialized module template is able to execute it’s blocks for all detector lines.

#### 6.2.3. TEMPLATE ARGUMENTS

Declaration of the module:

```
template<typename ... Args> class Module;
```

Template arguments are assumed to be of type : **Block**

#### 6.2.4. PUBLIC MEMBERS

**Function:**

```
int execute(const std::vector<std::string> &moduleInput, const std::vector<std::string> &moduleOutput, const std::string &globalConfigFilename, const std::string &localConfigFilename);
```

The function execute module blocks for all detector lines.

Arguments:

- `moduleInput` – module input data filenames
- `moduleOutput` – module output data filenames
- `globalConfigFilename` – a global configuration file name
- `localConfigFilename` – a local configuration file name

Module maps module input and output data filenames to block input and output filenames. For example:

The input to the Resampling block is : `reference_image_filename`, `geodetic_filename`

The input to the Atmosphere block is : `LUT_folder_path`, `sza_filename`, `oza_filename`, `raa_filename`, `geodetic_filename`

The result of the mapping – the input to the module is:

```
Reference_image_filename, geodetic_filename, LUTS_folder_path, sza_filename, oza_filename, raa_filename.
```

Note that `geodetic_filename` has only one occurance in input module file names. The mapping is module/blocks independent. Requirement for the blocks to use the same file id to map it to correct input/output filename.
Function : const IParamMapperBlock &getParamMapperBlock()const;

The function returns parameter mapper interface of type IParamMapperBlock. Important public members are:

printParams() : prints registered parameters identifiers for the module
getInputParamsNum() : returns registered input parameters number for the module
getOutputParamsNum() : returns registered output parameters number for the module
7. CREATING BLOCKS AND MODULES

This section presents to the user what is required to create a block.

7.1. CREATING NEW BLOCK

New block should inherit from Biblos.Block. It has to implement its virtual functions (see 6.1.4). As an example the Biblos.Orbit block is presented:

```cpp
namespace biblos
{
    namespace geometry
    {
        class B_Orbit : public biblos::Block<B_OrbitGlobalConfig, B_OrbitLocalConfig>
        {
            public:
                B_Orbit()
                : Block(std::make_pair(std::vector<std::string>({GeomFilenamesIds::iers_eop_time_init,
                    GeomFilenamesIds::ini_state_vector}),
                    std::vector<std::string>({GeomFilenamesIds::real_orbit,
                    GeomFilenamesIds::estimated_orbit})))
                {
                }
            virtual ~B_Orbit(){};
            int B_execute(const B_OrbitGlobalConfig &global_param, const B_OrbitLocalConfig &geom_param,
                const std::vector<std::string> &inputFiles, 
                const std::vector<std::string> &outputFiles) override;

            std::map<DetectorLineID, std::pair<B_OrbitGlobalConfig, B_OrbitLocalConfig>>
            parseConfigurationFiles(
                const std::string &globalConfigurationFilename, const std::string &
                localConfigurationFilename) const override;
            private:
                Logger B_Log;
        }
    }
}
```

The Orbit.Block implements the methods listed in 6.1.4. As a helper structure for file identifiers the Biblos.Orbit uses GeomFilenamesIds enumeration.

7.1.1. GETTING ARGUMENTS FROM INPUT/OUTPUT VECTOR

Input/output vector contains input and output data filenames. To get particular file using this file id passed to Block constructor, the user may use the following:

```cpp
std::string iersFilename = inputFiles[
    this->getRegisteredParamId(GeomFilenamesIds::iers_eop_time_init)
];
```

Function getRegisteredParamId provided by Biblos.Block allows the user to obtain file identification number in the vector using file identifier (here GeomFilenamesIds::iers_eop_time_init).
### 7.1.2. Parsing Configuration Files

For configuration files parsing the user may use helper functions implemented in `PropertyTreeHelper` class located in Biblos.Commons package. As an example Orbit block function for global configuration file parsing is presented:

```cpp
void B_OrbitGlobalConfig::read(const std::string &filename, B_OrbitGlobalConfig &config, const std::string &line)
{
    PropertyTreeHelper reader(filename);
    const auto & rootNode = reader.root();
    const auto & lineNode = reader.findParameterParentTree<std::string>(rootNode, "Detector_ID", line);
    config.detector_id = line;
    config.debug_mode = ReaderUtils::getDebugMode();
    config.instrument_sampling_time = reader.getParameterValue<double>(lineNode, "instrument_sampling_time");
    config.output_square_size = reader.getParameterValue<unsigned int>(lineNode, "output_square_size");
    config.satellite_position_attitude_sampling_time = reader.getParameterValue<double>(rootNode, "satellite_position_attitude_sampling_time");
}
```

### 7.2. Creating New Module (Connecting Blocks)

Each module should specialize Module template (see. 6.2). As an example SceneGeneration module implementation is presented:

```cpp
Module<B_Resampling, B_AtmosphereSimulatorMain> sceneGeneration;
int code = sceneGeneration.execute(moduleInput, moduleOutput, globalFilename_, localFilename_);
return code;
```

The one requirement is that template arguments (blocks) should inherit from Biblos.Block (see 6.1)
8. INSTALLATION GUIDELINES

This chapter presents Biblos installation guidelines: environment requirements that are common for all the blocks and modules as well as additional requirements for particular modules. For example installation commands for each package please refer to 9., for step-by-step guide refer to 10.

8.1. SYSTEM REQUIREMENTS

Libraries required by Biblos:

OSFI >= 3.1
EOCFI >= 4.9.1 cpp x64
boost >= 1.60.0
kissFFT >= 1.30 with headers installed in <kiss_installation_directory>/include/kiss_fft
netcdf >= 4.4.0
zlib >= 1.2.8
hdf5 >= 1.8.16
Eigen >= 3.2.8

Tools:

GCC toolchain >= 4.9
cmake >= 2.8.12
make


8.1.1. BLOCKS REQUIREMENTS

These requirements applies for:
Biblos.Orbit, Biblos.Attitude, Biblos.AOCSInstrumentCoupling, Biblos.SceneInteraction,
Biblos.Resampling, Biblos.Atmosphere, Biblos.Spatial and Biblos.Radiometric blocks

REQUIRED LIBRARIES:

Biblos.Commons >= 1.0

8.1.2. MODULES REQUIREMENTS

These requirements applies for:
Biblos.SceneGeneration, Biblos.Geometry and Biblos.Instrument

REQUIRED LIBRARIES:

Biblos.Commons >= 1.0

8.1.3. BIBLOS.GEOMETRY REQUIREMENTS

REQUIRED LIBRARIES:
- Biblos.Attitude >= 1.0
- Biblos.Orbit >= 1.0
- Biblos.AocsInstrumentCoupling >= 1.0
- Biblos.SceneInteraction >= 1.0

8.1.4. BIBLOS.SCENEGENERATOR REQUIREMENTS
- Biblos.Atmosphere >= 1.0
- Biblos.Resampling >= 1.0

8.1.5. BIBLOS.INSTRUMENT REQUIREMENTS
- Biblos.Spatial >= 1.0
- Biblos.Radiometric >= 1.0

8.2. FLAGS FOR BIBLOS INSTALLATION CMAKE

Recommended is installing all Biblos packages in one directory, e.g. /usr/local/biblos
Note that required COTS (Commercial Off the Shelf) libraries with installation script are available in Biblos.COTS package:
https://gmv-biblos.gmv.com/download/biblos-cots/ (see. 8.3).

If user decide to install all the Biblos packages in one directory, only REQUIRED flags should be provided.

REQUIRED:

pass Biblos block installation directory - here /usr/local/biblos : flag CMAKE_INSTALL_PREFIX
pass osfi installation directory - here /usr/local/openSF/OSFI : flag OSFI_INST_PATH
pass EOCFI installation directory - here /usr/local/EOCFI : flag EOCFI_INST_PATH
pass parent kissFFT installation directory - here /usr/local/biblos_cots/ : flag KISS_FFT_INST_PATH
pass parent netCDF installation directory - here /usr/local/biblos_cots/ : flag NETCDF_INST_PATH
pass parent Eigen installation directory - here /usr/local/biblos_cots/ : flag EIGEN_INST_PATH
pass parent Boost installation directory - here /usr/local/biblos_cots/ : flag BOOST_INST_PATH

NOT REQUIRED:
- DNO_ORB=1 do not compile orbit block
- DNO_ATT=1 do not compile attitude block
- DNO_AOC=1 do not compile aocs instrument coupling block
- DNO_SCE=1 do not compile scene interaction block
- DNO_ATM=1 do not compile atmospheric block
- DNO_RES=1 do not compile resampling blockq
- DNO_SPA=1 do not compile spatial block
- DNO_RAD=1 do not compile radiometric block
- DNO_GMOD=1 do not compile geometry module
- DNO_SMOD=1 do not compile scene generation module
-DNO_IMOD=1 do not compile instrument module

if you install all Biblos blocks in one directory these cmake FLAGS are NOT REQUIRED:
-DBIBLOS_INST_PATH=path biblos library installation directory path in case they were installed in different directory
-DBIBLOS_COMMONS_INST_PATH=path installation directory path of the biblos commons library in case the commons library was installed in different directory
-DBIBLOS_ORBIT_INST_PATH=path installation directory path of the biblos orbit block library in case the block was installed in different directory
-DBIBLOS_ATTITUDE_INST_PATH=path installation directory path of the biblos attitude block library in case the block was installed in different directory
-DBIBLOS_AOCS_INST_PATH=path installation directory path of the biblos aocs instrument coupling block library in case the block was installed in different directory
-DBIBLOS_SCENE_INTERACTION_INST_PATH=path installation directory path of the biblos scene interaction block library in case the block was installed in different directory
-DBIBLOS_ATMOSPHERE_INST_PATH=path installation directory path of the biblos atmosphere block library in case the block was installed in different directory
-DBIBLOS_RESAMPLING_INST_PATH=path installation directory path of the biblos resampling block library in case the block was installed in different directory
-DBIBLOS_SPATIAL_INST_PATH=path installation directory path of the biblos spatial block library in case the block was installed in different directory
-DBIBLOS_RADIOMETRIC_INST_PATH=path installation directory path of the biblos radiometric block library in case the block was installed in different directory

8.3. BIBLOS.COTS

Biblos.COTS is a package containing libraries used by Biblos libraries. Note that COTS “Commercial Off the Shelf” is used to follow the naming of the “OpenSF End-to-End Simulators Delivery, Installation, Running and Implementation Guidelines” (12/09/2012). Biblos.COTS only provide external libraries that are LGPL license compliant.

8.3.1. DESCRIPTION

Biblos.COTS is a package containing the following libraries:
- boost 1.60.0
- kissFFT 1.30
- netcdf 4.4.0
- zlib 1.2.8
- hdf5 1.8.16
- Eigen 3.2.8

Biblos.COTS contains all 3rd party libraries required by Biblos except from EOCFI and OSFI libraries. Installation script is located in biblos_cots directory (install_biblos_cots.sh). It requires 1 argument, which is Biblos.COTS installation directory, e.g. (usr/local/biblos_cots). Script requires root privileges. Example run command:
sudo sh install_biblos_cost.sh /usr/local/biblos_cots

8.4. TEST

All packages except Biblos.Commons contains test script and test data for running the block/module. Test script run_test.sh is located in data directory. To run the script set correct LD_LIBRARY_PATH and run the scripts with an argument: -b biblos_installation_directory, e.g. sh run_test.sh -b /usr/local/biblos
Package Biblos.POC contains single test script: test_All.sh

Possible test results:

success - OUTPUT FILES MATCH PATTERN. TEST RESULT: SUCCESS
execution failure - Error Occured. TEST RESULT: FAILURE
calculation failure - Files $1 and $2 are different. TEST RESULT: FAILURE
9. BIBLOS BLOCKS AND MODULES

This chapter presents Biblos blocks and modules user interface (input parameters) as well as example instructions to compile, install and run short test for each package. For step-by-step installation please refer to 10.

9.1. ORBIT BLOCK

9.1.1. CONFIGURATION

- Global configuration file name
- Local configuration file name

Note that installation packages does not contain block-specific configuration files, although each block has its own global and local configuration types and reading functions. Block configuration file can be specified based on B_OrbitLocalConfig and B_OrbitGlobalConfig structures.

9.1.2. INPUT

- Iers EOP initialization file name
- Initial state vector

9.1.3. OUTPUT

- Satellite orbit filename
- Restituted orbit filename

9.1.4. EXAMPLE COMPILATION AND INSTALLATION COMMANDS

- Installation directory: /usr/local/biblos
- EOCFI is assumed to be installed in /usr/local/EOCFI
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- COTS (Boost, Eigen, KissFFT, NetCDF, ZLIB, HDF5) are assumed to be installed in /usr/local/biblos_cots

```cmake```
    cmake -DCMAKE_INSTALL_PREFIX=/usr/local/biblos -DCMAKE_BUILD_TYPE=Release -DCMAKE_C_COMPILER=/usr/bin/gcc -DCMAKE_CXX_COMPILER=/usr/bin/g++
```

make

```bash```
    sudo make install
```

9.1.5. EXAMPLE RUN COMMAND

- Directories are relative to orbit/data directory
- Package is assumed to be installed in /usr/local/biblos.
- Biblos COTS are assumed to be installed in /usr/local/biblos_cots
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- EOCFI is assumed to be installed in /usr/local/EOCFI

```bash```
    export LD_LIBRARY_PATH=/usr/local/openSF/OSFI/lib:/usr/local/biblos_cots/lib:/usr/local/biblos_cots/lib64:
    /usr/local/EOCFI/libraries/LINUX64/:/usr/local/biblos/lib/
```

```
    /usr/local/biblos/bin/biblos_orbit input/B_GlobalConfig.xml,input/B_GeometryLocalConfig.xml,input/bulletinb-283.dat,input/B_Initial_State_Vector.xml out/orbit_sat,out/orbit_rest
```

9.1.6. EXAMPLE TEST COMMAND

- Assumed current directory: orbit/data
- Package is assumed to be installed in /usr/local/biblos.
- Biblos COTS are assumed to be installed in /usr/local/biblos_cots
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- EOCFI is assumed to be installed in /usr/local/EOCFI

export
LD_LIBRARY_PATH=/usr/local/openSF/OSFI/lib:/usr/local/biblos_cots/lib:/usr/local/biblos_cots/lib64/:
:/usr/local/EOCFI/libraries/LINUX64:/usr/local/biblos/lib/

sh run_test.sh -b /usr/local/biblos

9.2. ATTITUDE BLOCK

9.2.1. CONFIGURATION

- Global configuration file name
- Local configuration file name

Note that installation packages does not contain block-specific configuration files, although each block has its own global and local configuration types and reading functions. Block configuration file can be specified based on B_AttitudeLocalConfig and B_AttitudeGlobalConfig structures.

9.2.2. INPUT

- Iers EOP initialization file name
- Satellite orbit filename
- Restituted orbit filename

9.2.3. OUTPUT

- Nominal attitude file name
- Satellite attitude file name
- Restituted attitude file name

9.2.4. EXAMPLE COMPILATION AND INSTALLATION COMMANDS

- Installation directory: /usr/local/biblos
- EOCFI is assumed to be installed in /usr/local/EOCFI
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- COTS (Boost, Eigen, KissFFT, NetCDF, ZLIB, HDF5) are assumed to be installed in
  /usr/local/biblos_cots

cmake -DCMAKE_INSTALL_PREFIX=/usr/local/biblos -DCMAKE_BUILD_TYPE=Release -
- DOSFI_INSTALL_PATH=/usr/local/openSF/OSFI -DEOCFI_INSTALL_PATH=/usr/local/EOCFI -
- DKISS_FFT_INSTALL_PATH=/usr/local/biblos_cots -DNETCDF_INSTALL_PATH=/usr/local/biblos_cots -
- DBOOST_INSTALL_PATH=/usr/local/biblos_cots -DEIGEN_INSTALL_PATH=/usr/local/biblos_cots -
- DCMAKE_C_COMPILER=/usr/bin/gcc -DCMAKE_CXX_COMPILER=/usr/bin/g++

make

sudo make install
9.2.5. EXAMPLE RUN COMMAND

- Directories are relative to attitude/data directory.
- Package is assumed to be installed in /usr/local/biblos.
- Biblos COTS are assumed to be installed in /usr/local/biblos_cots.
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- EOCFI is assumed to be installed in /usr/local/EOCFI

```bash
export LD_LIBRARY_PATH=/usr/local/openSF/OSFI/lib:/usr/local/biblos_cots/lib:/usr/local/biblos_cots/lib64/:
/usr/local/EOCFI/libraries/LINUX64/:
/usr/local/biblos/lib/

/usr/local/biblos/bin/biblos_attitude input/B_GlobalConfig.xml,input/B_GeometryLocalConfig.xml
input/bulletinb-283.dat,expected/orbit_sat,expected/orbit_rest out/att_nom,out/att_sat,out/att_rest
```

9.2.6. EXAMPLE TEST COMMAND

- Assumed current directory: attitude/data
- Package is assumed to be installed in /usr/local/biblos.
- Biblos COTS are assumed to be installed in /usr/local/biblos_cots
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- EOCFI is assumed to be installed in /usr/local/EOCFI

```bash
export LD_LIBRARY_PATH=/usr/local/openSF/OSFI/lib:/usr/local/biblos_cots/lib:/usr/local/biblos_cots/lib64/:
/usr/local/EOCFI/libraries/LINUX64/:
/usr/local/biblos/lib/

sh run_test.sh -b /usr/local/biblos
```

9.3. AOCSINSTRUMENTCOUPLING BLOCK

9.3.1. CONFIGURATION

- Global configuration file name
- Local configuration file name

Note that installation packages does not contain block-specific configuration files, although each block has it's own global and local configuration types and reading functions. Block configuration file can be specified based on B_AOCSLocalConfig and B_AOCSGlobalConfig structures.

9.3.2. INPUT

- Satellite orbit file name
- Satellite attitude file name

9.3.3. OUTPUT

- LOS (Line of sight) file name
- Rotation matrix instrument to satellite file name
- Rotation matrix : orbit to ecef file name
- Rotation matrix : instrument to ecef file name
9.3.4. EXAMPLE COMPILATION AND INSTALLATION COMMANDS

- Installation directory: /usr/local/biblos
- EOCFI is assumed to be installed in /usr/local/EOCFI
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- COTS (Boost, Eigen, KissFFT, NetCDF, ZLIB, HDF5) are assumed to be installed in /usr/local/biblos_cots

```
cmake -DCMAKE_INSTALL_PREFIX=/usr/local/biblos -DCMAKE_BUILD_TYPE=Release -
DOSFI_INST_PATH=/usr/local/openSF/OSFI -DEOCFI_INST_PATH=/usr/local/EOCFI -
DKISS_FFT_INST_PATH=/usr/local/biblos_cots -DNETCDF_INST_PATH=/usr/local/biblos_cots -
DBOOST_INST_PATH=/usr/local/biblos_cots -DEIGEN_INST_PATH=/usr/local/biblos_cots -
DCMAKE_C_COMPILER=/usr/bin/gcc -DCMAKE_CXX_COMPILER=/usr/bin/g++
```

```
make
```

```
sudo make install
```

9.3.5. EXAMPLE RUN COMMAND

- Directories are relative to aocs/data directory.
- Package is assumed to be installed in /usr/local/biblos.
- Biblos COTS are assumed to be installed in /usr/local/biblos_cots.
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- EOCFI is assumed to be installed in /usr/local/EOCFI

```
export LD_LIBRARY_PATH=/usr/local/openSF/OSFI/lib:/usr/local/biblos_cots/lib:/usr/local/biblos_cots/lib64/:
/usr/local/EOCFI/libraries/LINUX64:/usr/local/biblos/lib/
```

```
/usr/local/biblos/bin/biblos_aocs input/B_GlobalConfig.xml,input/B_GeometryLocalConfig.xml
expected/orbit_sat,expected/att_sat out/los,out/roti2s,out/roto2e,out/roti2e
```

9.3.6. EXAMPLE TEST COMMAND

- Assumed current directory: aocs/data
- Package is assumed to be installed in /usr/local/biblos.
- Biblos COTS are assumed to be installed in /usr/local/biblos_cots
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- EOCFI is assumed to be installed in /usr/local/EOCFI

```
export LD_LIBRARY_PATH=/usr/local/openSF/OSFI/lib:/usr/local/biblos_cots/lib:/usr/local/biblos_cots/lib64/:
/usr/local/EOCFI/libraries/LINUX64:/usr/local/biblos/lib/
```

```
sh run_test.sh -b /usr/local/biblos
```

9.4. SCENEINTERACTION BLOCK

9.4.1. CONFIGURATION

- Global configuration file name
- Local configuration file name

Note that installation packages does not contain block-specific configuration files, although each block has its own global and local configuration types and reading functions. Block configuration file can be specified based on B_SceneLocalConfig and B_SceneGlobalConfig structures.
9.4.2. INPUT

- Iers EOP initialization file name
- Digital Elevation Model (DEM) file name
- Satellite orbit file name
- Rotation matrix: instrument to ecef file name
- LOS (Line of sight) file name

9.4.3. OUTPUT

- Geodetic file name
- OZA (Observation Zenith Angle) file name
- SZA (Solar Zenith Angle) file name
- RAA (Relative Azimuth Angle) file name

9.4.4. EXAMPLE COMPILATION AND INSTALLATION COMMANDS

- Installation directory: /usr/local/biblos
- EOCFI is assumed to be installed in /usr/local/EOCFI
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- COTS (Boost, Eigen, KissFFT, NetCDF, ZLIB, HDF5) are assumed to be installed in /usr/local/biblos_cots

```
cmake -DCMAKE_INSTALL_PREFIX=/usr/local/biblos -DCMAKE_BUILD_TYPE=Release -
DOSSI_INST_PATH=/usr/local/openSF/OSFI -DEOCFI_INST_PATH=/usr/local/EOCFI -
DKISS_FFT_INST_PATH=/usr/local/biblos_cots -DNETCDF_INST_PATH=/usr/local/biblos_cots -
DBOOST_INST_PATH=/usr/local/biblos_cots -DEIGEN_INST_PATH=/usr/local/biblos_cots -
DCMAKE_C_COMPILER=/usr/bin/gcc -DCMAKE_CXX_COMPILER=/usr/bin/g++
make
sudo make install
```

9.4.5. EXAMPLE RUN COMMAND

- Directories are relative to scene_interaction/data directory.
- Package is assumed to be installed in /usr/local/biblos.
- Biblos COTS are assumed to be installed in /usr/local/biblos_cots.
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- EOCFI is assumed to be installed in /usr/local/EOCFI

```
export
LD_LIBRARY_PATH=/usr/local/openSF/OSFI/lib:/usr/local/biblos_cots/lib:/usr/local/biblos_cots/lib64/
:/usr/local/EOCFI/libraries/LINUX64:/usr/local/biblos/lib/

/usr/local/biblos/bin/biblos_scene_interaction
input/B_GlobalConfig.xml,input/B_GeometryLocalConfig.xml input/bulletinb-283.dat,input/DEM_ACE29S_CFG.EOF,expected/orbit_sat,expected/roti2e,expected/los
out/geo,out/oza,out/sza,out/raa
```

9.4.6. EXAMPLE TEST COMMAND

- Assumed current directory: scene_interaction/data
- Package is assumed to be installed in /usr/local/biblos.
- Biblos COTS are assumed to be installed in /usr/local/biblos_cots
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- EOCFI is assumed to be installed in /usr/local/EOCFI
export
LD_LIBRARY_PATH=/usr/local/openSF/OSFI/lib:/usr/local/biblos_cots/lib:/usr/local/biblos_cots/lib64/:/usr/local/EOCFI/libraries/LINUX64/:/usr/local/biblos/lib/

sh run_test.sh -b /usr/local/biblos

9.5. RESAMPLING BLOCK

9.5.1. CONFIGURATION

- Global configuration file name
- Local configuration file name

Note that installation packages does not contain block-specific configuration files, although each block has its own global and local configuration types and reading functions. Block configuration file can be specified based on B_ResamplingLocalConfig and B_ResamplingGlobalConfig structures.

9.5.2. INPUT

- Top Of Canopy or psychical reference image
- Geodetic file name

9.5.3. OUTPUT

- Scene resampled file name

9.5.4. EXAMPLE COMPILATION AND INSTALLATION COMMANDS

- Installation directory: /usr/local/biblos
- EOCFI is assumed to be installed in /usr/local/EOCFI
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- COTS (Boost, Eigen, KissFFT, NetCDF, ZLIB, HDF5) are assumed to be installed in /usr/local/biblos_cots

```
make
sudo make install
```

9.5.5. EXAMPLE RUN COMMAND

- Directories are relative to resampling/data directory.
- Package is assumed to be installed in /usr/local/biblos.
- Biblos COTS are assumed to be installed in /usr/local/biblos_cots.
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- EOCFI is assumed to be installed in /usr/local/EOCFI

```
export
LD_LIBRARY_PATH=/usr/local/openSF/OSFI/lib:/usr/local/biblos_cots/lib:/usr/local/biblos_cots/lib64/:/usr/local/EOCFI/libraries/LINUX64/:/usr/local/biblos/lib/
```
9.5.6. EXAMPLE TEST COMMAND

- Assumed current directory: resampling/data
- Package is assumed to be installed in /usr/local/biblos.
- Biblos COTS are assumed to be installed in /usr/local/biblos_cots
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- EOCFI is assumed to be installed in /usr/local/EOCFI

export
LD_LIBRARY_PATH=/usr/local/openSF/OSFI/lib:/usr/local/biblos_cots/lib:/usr/local/biblos_cots/lib64:/usr/local/EOCFI/libraries/LINUX64:/usr/local/biblos/lib/

sh run_test.sh -b /usr/local/biblos

9.6. ATMOSPHERE BLOCK

9.6.1. CONFIGURATION

- Global configuration file name
- Local configuration file name

Note that installation packages does not contain block-specific configuration files, although each block has it's own global and local configuration types and reading functions. Block configuration file can be specified based on B_AtmoLocalConfig and B_AtmoGlobalConfig structures.

9.6.2. INPUT

- Look-Up-Tables folder with atmosphere data
- SZA (Solar Zenith Angle) file name
- OZA (Observation Zenith Angle) file name
- RAA (Relative Azimuth Angle) file name
- Geodetic file name

9.6.3. OUTPUT

- Top Of Atmosphere (TOA) file name

9.6.4. EXAMPLE COMPILATION AND INSTALLATION COMMANDS

- Installation directory: /usr/local/biblos
- EOCFI is assumed to be installed in /usr/local/EOCFI
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- COTS (Boost, Eigen, KissFFT, NetCDF, ZLIB, HDF5) are assumed to be installed in /usr/local/biblos_cots

```
make
sudo make install
```
9.6.5. EXAMPLE RUN COMMAND

- Directories are relative to atmosphere/data directory.
- Package is assumed to be installed in /usr/local/biblos.
- Biblos COTS are assumed to be installed in /usr/local/biblos_cots.
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- EOCFI is assumed to be installed in /usr/local/EOCFI

export LD_LIBRARY_PATH=/usr/local/openSF/OSFI/lib:/usr/local/biblos_cots/lib:/usr/local/biblos_cots/lib64:/usr/local/EOCFI/libraries/LINUX64:/usr/local/biblos/lib/

/atmosphere/data
input/B_GlobalConfig.xml,input/B_SceneGenerationLocalConfig.xml
input/LUTS/,expected/sza,expected/oza,expected/raa,expected/geo,expected/resm out/toa

9.6.6. EXAMPLE TEST COMMAND

- Assumed current directory: atmosphere/data
- Package is assumed to be installed in /usr/local/biblos.
- Biblos COTS are assumed to be installed in /usr/local/biblos_cots
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- EOCFI is assumed to be installed in /usr/local/EOCFI

export LD_LIBRARY_PATH=/usr/local/openSF/OSFI/lib:/usr/local/biblos_cots/lib:/usr/local/biblos_cots/lib64:/usr/local/EOCFI/libraries/LINUX64:/usr/local/biblos/lib/

sh run_test.sh -b /usr/local/biblos

9.7. SPATIAL BLOCK

9.7.1. CONFIGURATION

- Global configuration file name
- Local configuration file name

Note that installation packages does not contain block-specific configuration files, although each block has its own global and local configuration types and reading functions. Block configuration file can be specified based on B_SpatialLocalConfig and B_SpatialGlobalConfig structures.

9.7.2. INPUT

- Modulation Transfer Function (MTF) configuration file name
- Top Of Atmosphere (TOA) file name

9.7.3. OUTPUT

- Scene Irradiances

9.7.4. EXAMPLE COMPILATION AND INSTALLATION COMMANDS

- Installation directory: /usr/local/biblos
- EOCFI is assumed to be installed in /usr/local/EOCFI
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- COTS (Boost, Eigen, KissFFT, NetCDF, ZLIB, HDF5) are assumed to be installed in 
  /usr/local/biblos_cots

cmake -DCMAKE_INSTALL_PREFIX=/usr/local/biblos -DCMAKE_BUILD_TYPE=Release -
  DCMAKE_INSTALL_PREFIX=/usr/local/biblos -DCMAKE_BUILD_TYPE=Release -
  DOSFI_INST_PATH=/usr/local/openSF/OSFI -DEOCFI_INST_PATH=/usr/local/EOCFI -
  DKISS_FFT_INST_PATH=/usr/local/biblos_cots -DNETCDF_INST_PATH=/usr/local/biblos_cots -
  DBOOST_INST_PATH=/usr/local/biblos_cots -DEIGEN_INST_PATH=/usr/local/biblos_cots -
  DCMAKE_C_COMPILER=/usr/bin/gcc -DCMAKE_CXX_COMPILER=/usr/bin/g++

make
sudo make install

9.7.5. EXAMPLE RUN COMMAND

- Directories are relative to spatial/data directory.
- Package is assumed to be installed in /usr/local/biblos.
- Biblos COTS are assumed to be installed in /usr/local/biblos_cots.
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- EOCFI is assumed to be installed in /usr/local/EOCFI

export
LD_LIBRARY_PATH=/usr/local/openSF/OSFI/lib:/usr/local/biblos_cots/lib:/usr/local/biblos_cots/lib64/
:usr/local/EOCFI/libraries/LINUX64:/usr/local/biblos/lib/

/usr/local/biblos/bin/biblos_spatial input/B_GlobalConfig.xml,input/B_InstrumentLocalConfig.xml
input/MTF.xml,expected/toa out/scene_irr

9.7.6. EXAMPLE TEST COMMAND

- Assumed current directory: spatial/data
- Package is assumed to be installed in /usr/local/biblos.
- Biblos COTS are assumed to be installed in /usr/local/biblos_cots
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- EOCFI is assumed to be installed in /usr/local/EOCFI

export
LD_LIBRARY_PATH=/usr/local/openSF/OSFI/lib:/usr/local/biblos_cots/lib:/usr/local/biblos_cots/lib64/
:usr/local/EOCFI/libraries/LINUX64:/usr/local/biblos/lib/

sh run_test.sh -b /usr/local/biblos

9.8. RADIOMETRIC BLOCK

9.8.1. CONFIGURATION

- Global configuration file name
- Local configuration file name

Note that installation packages does not contain block-specific configuration files, although each block
has its own global and local configuration types and reading functions. Block configuration file can be
specified based on B_RadiometricGlobalConfig and B_RadiometricLocalConfig structures.

9.8.2. INPUT

- Scene Irradiances
9.8.3. OUTPUT

- Raw data image

9.8.4. EXAMPLE COMPILATION AND INSTALLATION COMMANDS

- Installation directory: /usr/local/biblos
- EOCFI is assumed to be installed in /usr/local/EOCFI
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- COTS (Boost, Eigen, KissFFT, NetCDF, ZLIB, HDF5) are assumed to be installed in /usr/local/biblos_cots


make

sudo make install

9.8.5. EXAMPLE RUN COMMAND

- Directories are relative to radiometric/data directory.
- Package is assumed to be installed in /usr/local/biblos.
- Biblos COTS are assumed to be installed in /usr/local/biblos_cots.
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- EOCFI is assumed to be installed in /usr/local/EOCFI

export

LD_LIBRARY_PATH=/usr/local/openSF/OSFI/lib:/usr/local/biblos_cots/lib:/usr/local/biblos_cots/lib64:/usr/local/EOCFI/libraries/LINUX64:/usr/local/biblos/lib

/users/local/biblos/bin/biblos_radiometric input/B_GlobalConfig.xml,input/B_InstrumentLocalConfig.xml

expected/scene_irr out/raw_data

9.8.6. EXAMPLE TEST COMMAND

- Assumed current directory: radiometric/data
- Package is assumed to be installed in /usr/local/biblos.
- Biblos COTS are assumed to be installed in /usr/local/biblos_cots
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- EOCFI is assumed to be installed in /usr/local/EOCFI

export

LD_LIBRARY_PATH=/usr/local/openSF/OSFI/lib:/usr/local/biblos_cots/lib:/usr/local/biblos_cots/lib64:/usr/local/EOCFI/libraries/LINUX64:/usr/local/biblos/lib

sh run_test.sh -b /usr/local/biblos

9.9. GEOMETRY MODULE

9.9.1. CONFIGURATION

- Global configuration file name
- Local configuration file name
9.9.2. INPUT

- Iers EOP initialization file name
- Initial state vector

9.9.3. OUTPUT

- Satellite orbit filename
- Restituted orbit filename
- Nominal attitude file name
- Satellite attitude file name
- Restituted attitude file name
- LOS (Line of sight) file name
- Rotation matrix instrument to satellite file name
- Rotation matrix : orbit to ecef file name
- Rotation matrix : instrument to ecef file name
- Geodetic file name
- OZA (Observation Zenith Angle) file name
- SZA (Solar Zenith Angle) file name
- RAA (Relative Azimuth Angle) file name

9.9.4. EXAMPLE COMPILATION AND INSTALLATION COMMANDS

- Installation directory: /usr/local/biblos
- EOCFI is assumed to be installed in /usr/local/EOCFI
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- COTS (Boost, Eigen, KissFFT, NetCDF, ZLIB, HDF5) are assumed to be installed in /usr/local/biblos_cots
- Biblos.Orbit, Biblos.Attitude, Biblos.AOCSInstrumentCoupling, Biblos.SceneInteraction are assumed to be installed in /usr/local/biblos
- Biblos_orbit, Biblos_attitude, Biblos_AOCSInstrumentCoupling, Biblos_SceneInteraction
- make
- sudo make install

9.9.5. EXAMPLE RUN COMMAND

- Directories are relative to geometry_module/data directory.
- Package is assumed to be installed in /usr/local/biblos.
- Biblos COTS are assumed to be installed in /usr/local/biblos_cots.
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- EOCFI is assumed to be installed in /usr/local/EOCFI
- export
- LD_LIBRARY_PATH=/usr/local/openSF/OSFI/lib:/usr/local/biblos_cots/lib:/usr/local/biblos_cots/lib64:/usr/local/EOCFI/libraries/LINUX64:/usr/local/biblos/lib/
- /usr/local/biblos/bin/geometry_module input/B_GlobalConfig.xml,input/B_GeometryLocalConfig.xml,input/bulletinb-283.dat,input/B_Initial_State_Vector.xml,input/DEM_ACE29S_CFG.EOF
9.9.6. EXAMPLE TEST COMMAND

- Assumed current directory: geometry_module/data
- Package is assumed to be installed in /usr/local/biblos.
- Biblos COTS are assumed to be installed in /usr/local/biblos_cots
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- EOCFI is assumed to be installed in /usr/local/EOCFI

export
LD_LIBRARY_PATH=/usr/local/openSF/OSFI/lib:/usr/local/biblos_cots/lib:/usr/local/biblos_cots/lib64:/usr/local/EOCFI/libraries/LINUX64:/usr/local/biblos/lib/

sh run_test.sh -b /usr/local/biblos

9.10. SCENEGENERATOR MODULE

9.10.1. CONFIGURATION

- Global configuration file name
- Local configuration file name

9.10.2. INPUT

- Top Of Canopy or physical reference image
- Geodetic file name
- Look-Up-Tables folder with atmosphere data
- SZA (Solar Zenith Angle) file name
- OZA (Observation Zenith Angle) file name
- RAA (Relative Azimuth Angle) file name

9.10.3. OUTPUT

- Scene resampled file name
- Top Of Atmosphere (TOA) file name

9.10.4. EXAMPLE COMPILATION AND INSTALLATION COMMANDS

- Installation directory: /usr/local/biblos
- EOCFI is assumed to be installed in /usr/local/EOCFI
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- COTS (Boost, Eigen, KissFFT, NetCDF, ZLIB, HDF5) are assumed to be installed in /usr/local/biblos_cots
- Biblos.Resampling and Biblos.Atmosphere are assumed to be installed in /usr/local/biblos

cmake -DCMAKE_INSTALL_PREFIX=/usr/local/biblos -DCMAKE_BUILD_TYPE=Release -
DOSFI_INST_PATH=/usr/local/openSF/OSFI -DEOCFI_INST_PATH=/usr/local/EOCFI -
DKISS_FFT_INST_PATH=/usr/local/biblos_cots -DNETCDF_INST_PATH=/usr/local/biblos_cots -
DBOOST_INST_PATH=/usr/local/biblos_cots -DEIGEN_INST_PATH=/usr/local/biblos_cots -
DCMAKE_C_COMPILER=/usr/bin/gcc -DCMAKE_CXX_COMPILER=/usr/bin/g++
make
sudo make install
**9.10.5. EXAMPLE RUN COMMAND**

- Directories are relative to scene_generation_module/data directory.
- Package is assumed to be installed in /usr/local/biblos.
- Biblos COTS are assumed to be installed in /usr/local/biblos_cots.
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- EOCFI is assumed to be installed in /usr/local/EOCFI

export
LD_LIBRARY_PATH=/usr/local/openSF/OSFI/lib:/usr/local/biblos_cots/lib:/usr/local/biblos_cots/lib64:/usr/local/EOCFI/libraries/LINUX64:/usr/local/biblos/lib/

/usr/local/biblos/bin/scenegen_module
input/B_GlobalConfig.xml,input/B_SceneGenerationLocalConfig.xml
input/ref.nc,expected/geo,input/LUTS/,expected/sza,expected/oza,expected/raa out/resm,out/toa

**9.10.6. EXAMPLE TEST COMMAND**

- Assumed current directory: scene_generation_module/data
- Package is assumed to be installed in /usr/local/biblos.
- Biblos COTS are assumed to be installed in /usr/local/biblos_cots
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- EOCFI is assumed to be installed in /usr/local/EOCFI

export
LD_LIBRARY_PATH=/usr/local/openSF/OSFI/lib:/usr/local/biblos_cots/lib:/usr/local/biblos_cots/lib64:/usr/local/EOCFI/libraries/LINUX64:/usr/local/biblos/lib/

sh run_test.sh -b /usr/local/biblos

**9.11. INSTRUMENT MODULE**

**9.11.1. CONFIGURATION**

- Global configuration file name
- Local configuration file name

**9.11.2. INPUT**

- Modulation Transfer Function (MTF) configuration file name
- Top Of Atmosphere (TOA) file name
- Scene Irradiances

**9.11.3. OUTPUT**

- Scene Irradiances
- Raw data image

**9.11.4. EXAMPLE COMPILATION AND INSTALLATION COMMANDS**

- Installation directory: /usr/local/biblos
- EOCFI is assumed to be installed in /usr/local/EOCFI
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- COTS (Boost, Eigen, KissFFT, NetCDF, ZLIB, HDF5) are assumed to be installed in /usr/local/biblos_cots
- Biblos.Spatial and Biblos.Radiometric are assumed to be installed in /usr/local/biblos
cmake -DCMAKE_INSTALL_PREFIX=/usr/local/biblos -DCMAKE_BUILD_TYPE=Release -
DOSFI_INST_PATH=/usr/local/openSF/OSFI -DEOFCI_INST_PATH=/usr/local/EOCFI -
DKISS_FFT_INST_PATH=/usr/local/biblos_cots -DNETCDF_INST_PATH=/usr/local/biblos_cots -
DBOOST_INST_PATH=/usr/local/biblos_cots -DEIGEN_INST_PATH=/usr/local/biblos_cots -
DCMAKE_C_COMPILER=/usr/bin/gcc -DCMAKE_CXX_COMPILER=/usr/bin/g++
make
sudo make install

9.11.5. EXAMPLE RUN COMMAND

- Directories are relative to instrument/data directory.
- Package is assumed to be installed in /usr/local/biblos.
- Biblos COTS are assumed to be installed in /usr/local/biblos_cots.
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- EOCFI is assumed to be installed in /usr/local/EOCFI
export
LD_LIBRARY_PATH=/usr/local/openSF/OSFI/lib:/usr/local/biblos_cots/lib:/usr/local/biblos_cots/lib64/
:/usr/local/EOCFI/libraries/LINUX64:/usr/local/biblos/lib/

/usr/local/biblos/bin/instrument_module input/B_GlobalConfig.xml,input/B_InstrumentLocalConfig.xml
input/MTF.xml,expected/toa out/scene_irr,out/raw_data

9.11.6. EXAMPLE TEST COMMAND

- Assumed current directory: instrument/data
- Package is assumed to be installed in /usr/local/biblos.
- Biblos COTS are assumed to be installed in /usr/local/biblos_cots
- OSFI is assumed to be installed in /usr/local/openSF/OSFI
- EOCFI is assumed to be installed in /usr/local/EOCFI
export
LD_LIBRARY_PATH=/usr/local/openSF/OSFI/lib:/usr/local/biblos_cots/lib:/usr/local/biblos_cots/lib64/
:/usr/local/EOCFI/libraries/LINUX64:/usr/local/biblos/lib/

sh run_test.sh -b /usr/local/biblos
10. STEP-BY-STEP INSTALLATION GUIDE

10.1. REQUIREMENTS FOR STEP-BY-STEP GUIDE

1. Operating system: OpenSUSE 13.2
2. Created user account eomodel with home directory: /home/eomodel
3. Installed openSF 3.4 with OSFI, database user: openSF, database password: openSF
   Installed in /usr/local (/usr/local/openSF, /usr/local/openSF/OSFI)
4. Installed EOCFI cxx 64 bit 4.9.1, installation directory: /usr/local (/usr/local/EOCFI)
5. Installed zip, GNU m4, unzip
6. Start logged as eomodel user in /home/eomodel directory.

10.2. STEPS

10.2.1. INSTALLATION OF COTS

<table>
<thead>
<tr>
<th>Action</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Click on the download button and save file to /home/eomodel/biblos_cots.zip</td>
<td></td>
</tr>
<tr>
<td>3. Unzip the package</td>
<td>unzip biblos_cots.zip</td>
</tr>
<tr>
<td>4. Go into biblos_cots directory</td>
<td>cd biblos_cots</td>
</tr>
<tr>
<td>5. Execute installation script</td>
<td>sudo sh install_biblos_cots.sh /usr/local/biblos_cots</td>
</tr>
<tr>
<td>6. Wait for the installation process to finish. Check if the last message is correct: *** FINISHED INSTALLATION OF COTS ***</td>
<td></td>
</tr>
<tr>
<td>7. Go into /home/eomodel directory</td>
<td>cd ..</td>
</tr>
</tbody>
</table>

Table 10-1: Installation of COTS

10.2.2. INSTALLATION OF BIBLOS LIBRARIES

Requires previous step (see 10.2.1).

<table>
<thead>
<tr>
<th>Action</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>9. Click on the download button and save file to /home/eomodel/ biblos_software.zip</td>
<td></td>
</tr>
<tr>
<td>10. Unzip the package</td>
<td>unzip biblos_software.zip</td>
</tr>
<tr>
<td>11. Go into biblos_software directory</td>
<td>cd biblos_software</td>
</tr>
</tbody>
</table>
12. Execute installation script
   
   Command: `sudo sh install_biblos.sh -i /usr/local/biblos`

13. Provide paths for libraries:
   - please provide NETCDF installation directory: `/usr/local/biblos_cots/`
   - please provide EIGEN installation directory: `/usr/local/biblos_cots/`
   - please provide KISSFFT installation directory: `/usr/local/biblos_cots/`
   - please provide BOOST installation directory: `/usr/local/biblos_cots/`

   Command:
   ```
   /usr/local/biblos_cots
   /usr/local/biblos_cots
   /usr/local/biblos_cots
   /usr/local/biblos_cots
   ```

14. Wait for the installation process to finish. Check if the last message is correct:
   Biblos installed succesfully

   *** FINISHED INSTALLATION OF BIBLOS ***

15. Export LD_LIBRARY_PATH
   
   Command: `export LD_LIBRARY_PATH=/usr/local/openSF/OSFI/lib:/usr/local/biblos_cots/lib:/usr/local/biblos_cots/lib64:/usr/local/EOCFI/libraries/LINUX64:/usr/local/biblos/lib/`

16. Execute testAll script
   
   Command: `sh testAll.sh -b /usr/local/biblos`

17. Check if result is correct:
   OUTPUT FILES MATCH PATTERN. TEST RESULT: SUCCESS

18. Go into `/home/eomodel` directory
   
   Command: `cd ..`

### Table 10-2: Installation of all Biblos libraries

<table>
<thead>
<tr>
<th>Action</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>12. Execute installation script</td>
<td><code>sudo sh install_biblos.sh -i /usr/local/biblos</code></td>
</tr>
<tr>
<td>13. Provide paths for libraries:</td>
<td><code>/usr/local/biblos_cots/</code></td>
</tr>
<tr>
<td>14. Wait for the installation process to finish. Check if the last message is correct:</td>
<td>Biblos installed successfully</td>
</tr>
<tr>
<td>15. Export LD_LIBRARY_PATH</td>
<td><code>export LD_LIBRARY_PATH=/usr/local/openSF/OSFI/lib:/usr/local/biblos_cots/lib:/usr/local/biblos_cots/lib64:/usr/local/EOCFI/libraries/LINUX64:/usr/local/biblos/lib/</code></td>
</tr>
<tr>
<td>16. Execute testAll script</td>
<td><code>sh testAll.sh -b /usr/local/biblos</code></td>
</tr>
<tr>
<td>17. Check if result is correct:</td>
<td>OUTPUT FILES MATCH PATTERN. TEST RESULT: SUCCESS</td>
</tr>
<tr>
<td>18. Go into <code>/home/eomodel</code> directory</td>
<td><code>cd ..</code></td>
</tr>
</tbody>
</table>

10.2.3. INTEGRATION OF EXAMPLE SCENARIO AND POC WITH OPENSF, RUNNING EXAMPLE AND POC IN OPENSF

Requires previous step (see. 10.2.2)

<table>
<thead>
<tr>
<th>Action</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>19. Go into biblos_software directory</td>
<td><code>cd biblos_software</code></td>
</tr>
<tr>
<td>20. Load database biblos_poc</td>
<td><code>mysql --user=openSF --password=openSF &lt; biblos_poc.mysql</code></td>
</tr>
<tr>
<td>21. Add database to OpenSF</td>
<td><code>sudo echo &quot;db.biblos_poc=openSF;openSF;localhost&quot; &gt;&gt; /usr/local/openSF/openSF.properties</code></td>
</tr>
</tbody>
</table>

### Table 10-3: Integration of POC and example scenario database with OpenSF

To run example scenario:

<table>
<thead>
<tr>
<th>Action</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>22. Run openSF</td>
<td><code>openSF</code></td>
</tr>
<tr>
<td>Action</td>
<td>Command</td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| 23. Update LD_LIBRARY_PATH: | Ctrl+O  
Open configuration:  
Add to LD_LIBRARY_PATH:  
:/usr/local/biblos_cots/lib:/usr/local/biblos/lib:/usr/local/EOCFI/libraries/LINUX64:/ |

| 24. Connect to biblos_poc database | Go into System->Databases  
Pick biblos_poc  
Click connect  
Click ok |

| 25. Run session GSI_example | Unwind sessions  
Right click on GSI_example  
In context menu: left click Run |

Table 10-4: Running example scenario in OpenSF

To run POC:

<table>
<thead>
<tr>
<th>Action</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>26. Run openSF</td>
<td>openSF</td>
</tr>
</tbody>
</table>

| 27. Update LD_LIBRARY_PATH: | Ctrl+O  
Open configuration:  
Add to LD_LIBRARY_PATH:  
:/usr/local/biblos_cots/lib:/usr/local/biblos/lib:/usr/local/EOCFI/libraries/LINUX64:/ |

| 28. Run session GSI_POC | Unwind sessions  
Right click on GSI_POC  
In context menu: left click Run |

Table 10-5: Running POC in OpenSF
END OF THE DOCUMENT