EUMETSAT

EUMETSAT WP FCIDECOMP - Solution design

EUMETSAT

Nov 10, 2021

LIST OF TABLES

CONTENTS:

CHAPTER

ONE

DOCUMENT INFORMATION

ID	:	fcidecomp/documentation/solution-design
Version	:	
Authors	:	M. Bottaccio (B-Open Solutions) M. Cucchi (B-Open Solutions)

1.1 Document Change Record

Table 1.1: Document Change Record

Issue / Revision	Date	DCN. No	Changed Pages / Paragraphs
	15 Nov 2021		First version.

CHAPTER

TWO

INTRODUCTION

2.1 Purpose

The document describes a design proposal for a maintainable solution allowing users to reliably decode FCI L1c products compressed with CharLS.

2.2 Reference Documents

Table 2.1: Reference documents

#	Title	Reference
[CONDA_VARIANTS]	conda-build – Build variants	https://docs.conda.io/projects/conda-build/en/latest/resources/variants.html
[FCIDECOMP_CONDA]	FCIDECOMP Conda recipe developed by Martin Raspaud (SMHI)	https://github.com/mraspaud/fcidecomp-conda-recipe/
[FCIDECOMP_LATEST]	FCIDECOMP v1.0.2 repository	https://sftp.eumetsat.int/public/folder/UsCVknVOOkSyCdgpMimJNQ/User-Materials/Test-Data/MTG/MTG_FCI_L1C_Enhanced-NonN_TD-272_May2020/FCI_Decompression_Software_V1.0.2/EUMETSAT-FCIDECOMP_V1.0.2.tar.gz
[FCIDECOMP_TEST_DATA]	FCIDECOMP v1.0.2 test data	https://sftp.eumetsat.int/public/folder/UsCVknVOOkSyCdgpMimJNQ/User-Materials/Test-Data/MTG/MTG_FCI_L1C_Enhanced-NonN_TD-272_May2020/
[FCIDECOMP_WPD]	Work Package Description	EUM/SEP/WPD/21/1244304
[HDF5PLUGIN]	`hdf5plugin` python package	https://github.com/silx-kit/hdf5plugin
[HDFVIEW]	HDFView Software	https://www.hdfgroup.org/downloads/hdfview/
[MTG4AFRICA]	EUMETSAT Data Tailor mtg4africa plugin	https://gitlab.eumetsat.int/data-tailor/support-to-mtg/mtg4africa
[NETCDF_C]	Unidata - NetCDF-C	https://docs.unidata.ucar.edu/netcdf-c/current/
[NETCDF_JAVA]	Unidata - NetCDF-Java	https://www.unidata.ucar.edu/software/netcdf-java/
[NETCDF_JAVA_GITHUB1]	NetCDF-C for reading (nj22Config.xml) in non-Unidata netCDF-Java based tools	https://github.com/Unidata/thredds/issues/1063

continues on next page

Table 2.1 – continued from previous page

#	Title	Reference
[PANOPLY]	Panoply netCDF, HDF and GRIB Data Viewer	https://www.giss.nasa.gov/tools/panoply/

CHAPTER

THREE

CREATION OF CANONICAL REPOSITORY

3.1 Introduction

A canonical repository is established on the EUMETSAT GitLab service at https://gitlab.eumetsat.int/sepdssme/fcidecomp for development purposes. Each time a new release is produced, the corresponding code is synchronized to the public EUMETSAT Open Source repository at **XXX [NOTE: abbiamo una reference?]**.

3.2 Repository initialization

FCIDECOMP v1.0.2 is taken as blueprint for the development of the solution codebase.

The repository is put under configuration control. A new minor release adding README, BUILD, INSTALL, and LICENCE files, starting the Changelog, codifying the use of semantic versioning for future versions and adding a standardised build system is published.

3.3 Test suite

An initial test suite (at least against nominal conditions) is implemented following the V&V strategy defined in **[TODO: add reference]**. Most tests are implemented as automated tests against the Python interface.

3.4 Test data

A preliminary set of test data taken from the *MTG FCI L1C test data* is added to ensure a consistent and permanent dataset to execute tests.

CHAPTER

FOUR

SUPPORT TO REQUIRED USAGE PATTERNS

4.1 Introduction

This section describes the strategies adopted to ensure that the FCIDECOMP software supports the required usage patterns.

[NOTA: questo l'ho spostato qui da Packaging and deployment]

As a baseline, the FCIDEOCOMP software supports HDF5 1.10. Strategies to grant support for multiple versions of HDF5 described in the *Further developments appendix <further_developments*.

4.2 Integration with tools based on netCDF-C

[NOTA: mi sembra che questa parte stia meglio in un ulteriore sottoparagrafo invece che nell'intro, ma se non sei d'accordo la sposto su e tolgo questo paragrafo]

The current implementation of the FCIDECOMP software (*v1.0.2*) which, as mentioned in the *Repository initialization* paragraph serves as blueprint for the software codebase, already satisfies the HDF5 filters interface. Given this, integration with utilities relying on the `netcdf-c` library (*[NETCDF-C]*) is ensured, provided that:

- the location of the FCIDECOMP filter library is specified in a specific environment variable, `HDF5_PLUGIN_PATH`;
- the correct filter id (32018 for FCIDECOMP), if required by the utility, is specified;

4.3 Usage as CLI tool

In order to provide a baseline support for CLI usage of the FCIDECOMP software, `nccopy` (software utility of the `netcdf-c` library) is chosen as reference standard CLI tool. To foster integration with `nccopy`, the FCIDECOMP software provides to:

- put the filter's library to a specific path at installation
- have the `HDF5_PLUGIN_PATH` environment variable automatically set each time a conda environment where FCIDECOMP is installed get activated

The FCIDECOMP software documentation also provides instructions on how to call `nccopy` to decompress files using the FCIDECOMP filter.

4.4 Integration with Python

Integration with Python is provided by a small Python package developed ad hoc, which satisfies the required `h5py` interface to make the FCIDECOMP filter available for Python applications. Such package, based upon a stripped-down version of the *hdf5plugin* package, is essentially composed of an `__init__.py` defining the filter interface to `h5py`.

See the *Integration with hdf5plugin appendix* for details on the integration with the widely used *hdf5plugin* package and interation with its maintainers' community.

4.5 Integration with EUMETSAT Data Tailor

At the moment, the Data Tailor supports reading compressed FCI L1C products through the optional `epct_plugin_mtg4africa` *customisation plugin*, which in turns install FCIDECOMP by installing with `pip` the `hdf5plugin` package.

The approach to integrate the described solution with the Data Tailor includes a revision of the current build and installation approach for the `epct_plugin_mtg4africa` customisation plugin, so that it installs the FCIDECOMP support through the Python package described above and its dependencies.

4.6 Integration with tools based on netCDF-Java

Panoply and *HDFView* have been identified as the key software based on netCDF-Java to support. The integration of the FCIDECOMP software in these applications can be achieved by instructing them to use the netCDF-C library (instead of netCDF-Java) to read netCDF files (see related *github issue*). Support is then granted by describing the aforementioned procedure in the FCIDECOMP software documentation.

The issue of a generic integration with *Unidata Netcdf-Java* is discussed in the *Design justification appendix*.

CHAPTER

FIVE

PACKAGING AND DEPLOYMENT

5.1 Introduction

In the following paragraphs the strategy to build and package the FCIDECOMP software in order to ensure support for all the required systems is reported.

5.2 Supported platforms

The FCIDECOMP software supports the following platforms:

- Windows 10, 32 and 64 bit
- Ubuntu 18.04, Ubuntu 20.04 64 bit
- CentOS 7 64 bit

Details on the selection process leading to the list presented above are provided in the *Design justification appendix*.

5.3 Building the binaries

The build system for the software binaries is drawn from the one used in the *FCIDECOMP v1.0.2 source code*, and adapted from there to guarantee support for all the required systems.

5.4 Packaging as a Conda package

Packages are built using Conda, as it provides standardised environments with a large set of pre-compiled packages. From the point of view of Conda, the operating systems listed in the *Supported platforms* paragraph can be considered as two groups of OS: in Conda standardised environment it is enough to build the package for one Linux distribution in order to make it compatible with other Linux distributions. So two conda packages are released: one for Linux distributions, and one for Windows 10.

These conda packages install both the FCIDECOMP libraries and its Python bindings. As a blueprint for the conda recipes, the *Conda recipe* for the packaging of FCIDECOMP mantained by Martin Raspaud from the Swedish Meteorological and Hydrological Institute has been used.

Conda packages are uploaded to EUMETSAT Anaconda repository **[NOTA: abbiamo una reference?]**.

5.5 Packaging process

GitLab CI/CD pipelines to compile, build, test and upload the conda packages to EUMETSAT Anaconda repository are implemented.

Two GitLab runners are implemented, one with a Docker executor on Linux and the other with a Shell executor on Windows. **[NOTA: se va presentata come una cosa già fatta, come inserire che non siamo sicure se serva un altro runner per Windows 32-bit?]**

CHAPTER

SIX

APPENDIX - TRACEABILITY MATRIX

CHAPTER

SEVEN

APPENDIX - LIST OF USERS AND DEVELOPERS CURRENTLY USING FCIDECOMP

CHAPTER

EIGHT

APPENDIX - DESIGN JUSTIFICATIONS

CHAPTER

NINE

APPENDIX - LONG-TERM PRESERVATION OF DEPENDENCIES

CHAPTER

TEN

APPENDIX - INTEGRATION WITH `HDF5PLUGIN`

CHAPTER

ELEVEN

APPENDIX - FURTHER DEVELOPMENTS