

USER GUIDE OF THE MTG FCI L1C DECOMPRESSION SOFTWARE

Doc.No. : EUM/MTG/MAN/16/858107
Issue : v1 e-signed
Date : 13 June 2016
WBS/DBS :

EUMETSAT
Eumetsat-Allee 1, D-64295 Darmstadt, Germany
Tel: +49 6151 807-7
Fax: +49 6151 807 555
<http://www.eumetsat.int>

Document Change Record

<i>Issue / Revision</i>	<i>Date</i>	<i>DCN. No</i>	<i>Changed Pages / Paragraphs</i>
1.0	13-06-2016	-	First version

Table of Contents

1	INTRODUCTION.....	5
1.1	Purpose	5
1.2	Scope	5
1.3	Applicable Documents	5
1.4	Reference Documents	5
1.5	Document Structure	5
2	FCI DECOMPRESSION SOFTWARE USER GUIDE	6
2.1	Purpose of the software	6
2.2	Operations environment.....	6
2.2.1	General	6
2.2.2	Hardware configuration.....	7
2.2.3	Software configuration	7
2.2.4	Operational constraints	9
2.3	Operations manual.....	9
2.3.1	Set-up and initialisation.....	9
2.3.2	Getting started.....	9
2.3.3	Normal operations.....	9
2.3.4	Normal termination.....	11
2.3.5	Error conditions	11
2.4	Reference manual.....	12
2.4.1	Introduction	12
2.4.2	Help method.....	12
2.4.2.1	User application makes use of the JAVA implementation of netCDF	12
2.4.2.2	HDF5 cannot find the HDF5 JPEG-LS filter plugin	12
2.4.2.3	CharLS library cannot be found on the system	13
2.4.3	Commands and operations.....	13
2.4.4	Error messages.....	13
2.4.4.1	Messages from the decompression filter: H5Zjpegls	13
2.4.4.2	Messages from the façade to CharLS: fcicomp_jpegls.....	13
2.5	Tutorial	15
2.5.1	Introduction	15
2.5.2	Getting started.....	15
2.5.3	Using the software on typical tasks	16
2.5.3.1	Examining the contents of a netCDF-4 variable and access to its compression ratio.....	16
2.5.3.2	Copying the compressed netCDF-4 file enabling deflate compression instead of JPEG-LS compression.....	17
2.5.3.3	Displaying the effective radiance data	17

Table of Figures

Figure 1 :	Modules in the FCI decompression software	7
Figure 2:	Libraries and links in the FCI decompression software	16
Figure 3:	Screenshot of ncview showing the effective radiance data of channel VIS 0.9	18

1 INTRODUCTION

This document is the Software User Guide (SUG) of the MTG FCI L1c decompression software.

1.1 Purpose

The purpose of this document is to provide instructions for the users of the software. It describes the usage of the FCI L1c decompression software. It provides information on how to access data from compressed netCDF datasets.

1.2 Scope

This document may interest the users of the FCI L1c decompression software, in particular developers of FCI Test Datasets, the developers and testers of systems which read FCI L1c data and End-Users.

1.3 Applicable Documents

[SIG] Software Installation Guide of the MTG FCI L1c decompression software EUM/MTG/MAN/16/858121

1.4 Reference Documents

[RD1] <https://www.unidata.ucar.edu/software/netcdf/docs/index.html> The NetCDF Documentation

1.5 Document Structure

This document is organised as follows:

- Section 1 provides the scope and content of the document;
- Section 2 is the FCI L1c Decompression Software User Guide:
 - Section 2.1 describes the purpose of the FCI L1c decompression software.
 - Section 2.2 describes the operational environment in which it is foreseen to run the software.
 - Section 2.3 is the operations manual. It describes how the software can be operated.
 - Section 2.4 is the reference manual. It provides useful information on the software usage.
 - Section 2.5 is a tutorial. It describes how to use software.

2 FCI DECOMPRESSION SOFTWARE USER GUIDE

2.1 Purpose of the software

The purpose of the FCI decompression software is to decompress the data that have been compressed with the FCI compression software in netCDF-4 files. These data are namely the effective radiance and pixel quality flags in the FCI L1c datasets.

2.2 Operations environment

2.2.1 General

The FCI decompression software is composed of modules in beige that are shown in **Figure 1**.

It includes:

- The façade to CharLS (fcicomp-jpegls) which allows separating the core of the FCI decompression software from CharLS processing functions.
- The HDF5 decompression filter (fcicomp-H5Zjpegls) is the filter that is dynamically loaded by HDF5 to decompress the data that have been encoded in JPEG-LS HDF5 compression filter.

Modules in blue are COTS components that have to be installed on the system in order to use the FCI decompression software. They are:

- CharLS software: the optimized implementation of the JPEG-LS standard for lossless and near-lossless image compression;
- The HDF5 library which is the storage layer used by netCDF-4 to read data from disk and write data to disk.
- The standard netCDF-4 C library which is used for accessing datasets from files in the netCDF-4 format.

The installation guide of the FCI decompression software is provided in [SIG].

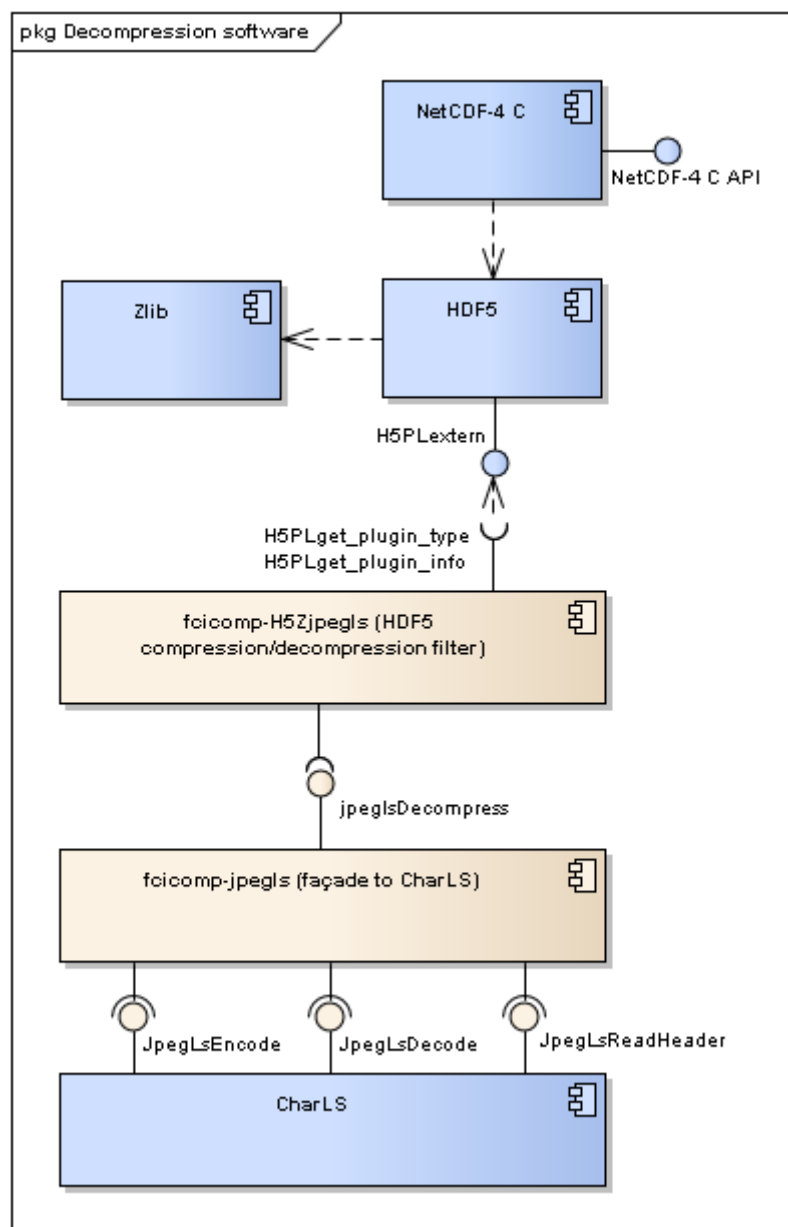


Figure 1 : Modules in the FCI decompression software

2.2.2 Hardware configuration

The FCI decompression software is known to work properly on the following hardware:

- PC Dell Precision T1650 equipped with 4 Intel Xeon CPU E3-1225 V2 @ 3.2GHz, 8MB and with 3.7GB 1600MHz DDR3 memory running CentOS release 6.3 (64 bits).

The FCI decompression software may work properly on less demanding hardware configuration.

(EUMETSAT may be able to later complement end-user minimum hardware specifications for the decompression, based on what is expected in end-user receiving stations).

2.2.3 Software configuration

The FCI decompression software is known to work properly on the following OS:

- Linux CentOS 6.3 x64 Edition

- Red Hat Enterprise Linux Client release 5.2 (Tikanga)
- Red Hat Enterprise Linux 6.5 Server x64 Edition

In order to build and install the FCI decompression software the following software need to be available on the system:

Tool	Minimum version	Purpose	Website
Gcc	4.4.6	Compiler for the FCI decompression software.	https://gcc.gnu.org
GNU make	3.81	Tool to build and install the FCI decompression software.	www.gnu.org/software/make
CMake	2.8.12	Tool to build the FCI decompression software.	www.cmake.org

The FCI decompression software is also linked to the following software that are also required to build and run it:

Tool	Minimum version	Purpose	Website
CharLS	1.0	Optimized implementation of the JPEG-LS standard for lossless and near-lossless image compression.	http://charls.codeplex.com/
Zlib	1.2.8	Lossless data-compression library that implement the "deflate" method. This library is required by HDF5.	http://www.zlib.net/
HDF5	1.8.14	Data model, library, and file format for storing and managing data. It is the storage layer used by netCDF-4. Note: The HDF5 library should have been configured with the --enable-hl option during its building process.	http://www.hdfgroup.org/HDF5/
netCDF-C	4.3.2	Set of software libraries and self-describing, machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data. Note: The netCDF-C library should have been configured with the --enable-netcdf-4 option during its building process.	http://www.unidata.ucar.edu/software/netcdf/

The following COTS are optional:

Tool	Recommended version	Purpose	Website
sha256sum	4.4.6	Compute and check SHA256 message digest. Sha256sum is part of the GNU coreutils package.	https://www.gnu.org/software/coreutils/
libcurl	7.19.7	Client-side URL transfer library. This library is required to build the DAP client of netCDF.	http://curl.haxx.se/

The following software could be used with FCI decompression software (optional):

Tool	Minimum version	Purpose	Website
ncview	2.1.2	Visual browser for netCDF files distributed with a GNU GPL licence.	http://meteora.ucsd.edu/~pierce/ncview_home_page.html

2.2.4 Operational constraints

In nominal situations, the FCI decompression software decompresses the data read from the compressed netCDF-4 file.

In non-nominal situations, i.e. if the decompression fails for any reason, the FCI decompression software prints an informative error to the standard error stream. See section 2.4.4 for a list of error messages.

2.3 Operations manual

2.3.1 Set-up and initialisation

The set-up and initialisation processes are described in the Software Installation Guide [SIG].

2.3.2 Getting started

Once correctly installed on the system, the FCI decompression software decompresses seamlessly the compressed data i.e. without any further intervention required by the user.

However, in order to be able to read the compressed data stored into the netCDF-4 files, the software tools accessing the netCDF-4 dataset have to be dynamically linked to the netCDF-4 C library. Other software tools (e.g. those including a java version of the netCDF library) will not be able to decode the compressed data.

Examples of software tools that can be dynamically linked to the netCDF-4 C library are the `ncdump` and `nccopy` utilities (which are part of the netCDF-C software package) or `ncview` visual browser for netCDF files (see section 2.5).

Problems users may encounter during the use of the FCI decompression software and the solution to these problems are provided in section 2.4.2. Error messages the FCI decompression software can return are listed in section 2.4.4.

2.3.3 Normal operations

This section provides the source code of a toy example that accesses the effective radiance variable and the pixel quality flag variable in the netCDF-4 file “simple_xy_nc4_jls.nc”. It shows that the data compressed with the FCI compression software are accessed seamlessly, i.e. without any explicit call to a decoding function, but making calls only to standard routines offered by the netCDF-4 C library interface.

```
#include <stdlib.h>
#include <stdio.h>
#include <netcdf.h>

/* This is the name of the data file we will read. */
#define FILENAME "simple_xy_nc4_jls.nc"

/* We are reading 2D data, a 30 x 60 grid. */
#define NX 30
```



```
#define NY 60

/* Handle errors by printing an error message and exiting with a
 * non-zero status. */
#define ERRCODE 2
#define ERR(e) {printf("Error: %s\n", nc_strerror(e)); exit(ERRCODE);}

int main() {

    /* Loop indexes, and error handling. */
    int x, y, retval;

    /* Open the file. NC_NOWRITE tells netCDF we want read-only access
     * to the file.*/
    int ncid = 0;
    if ((retval = nc_open(FILENAME, NC_NOWRITE, &ncid)))
        ERR(retval);

    /* Get the varid of the effective radiance variable, based on its name. */
    int varid = 0;
    if ((retval = nc_inq_varid(ncid, "Effective radiance", &varid)))
        ERR(retval);

    /* Initialize the effective radiance data matrix */
    short effective_radiance_data[NX][NY] = {{0}};
    /* Read the effective radiance data. */
    if ((retval = nc_get_var_short(ncid, varid,
&effective_radiance_data[0][0])))
        ERR(retval);

    /* Get the varid of the pixel quality flags variable, based on its name. */
    if ((retval = nc_inq_varid(ncid, "Pixel quality flags", &varid)))
        ERR(retval);

    /* Initialize the pixel quality flags matrix */
    unsigned char pixel_quality_flags[NX][NY] = {{0}};
    /* Read the pixel quality flags data. */
    if ((retval = nc_get_var_uchar(ncid, varid, &pixel_quality_flags[0][0])))
        ERR(retval);

    /* Check the data. */
    for (x = 0; x < NX; x++)
        for (y = 0; y < NY; y++) {
            /* Compare the radiance data to the data created in
simple_xy_nc4_jls_wr. */
            if (effective_radiance_data[x][y] != (short) (x * NY + y))
                return ERRCODE;
            /* Compare the pixel quality flag data to the data created in
simple_xy_nc4_jls_wr. */
            if (pixel_quality_flags[x][y] != (unsigned char) (x))
                return ERRCODE;
        }

    /* Close the file, freeing all resources. */
    if ((retval = nc_close(ncid)))
        ERR(retval);

    /* Print a success message and exit. */
}
```

```
printf("*** SUCCESS reading example file %s!\n", FILENAME);  
return 0;  
}
```

Other examples of usage with nccopy and ncview software are provided in section 2.5.

2.3.4 Normal termination

Normal termination depends on the application the user is running.

In case of normal termination the FCI decompression software should not have printed any error to the standard error stream and the data should have been decompressed successfully.

In this case, user application should continue without error.

For example, if the application is the sample code provided in section 2.3.3, the calls to the `nc_get_var_short` and `nc_get_var_uchar` functions return `NC_NOERR` (no error) and the program execution continues normally. If the application is `ncview`, the decompressed image is displayed on the screen.

2.3.5 Error conditions

In case an error detected by the FCI decompression software, the data are not decompressed and the user application may return an error.

For example, with the sample code provided in section 2.3.3, if FCI decompression software detects an error, the calls to the `nc_get_var_short` and `nc_get_var_uchar` functions return `NC_EHDFERR` (error reported by HDF5 layer) and the program exits with an error.

With `ncview`, if FCI decompression software detects an error, it returns an `NC_EHDFERR` and `ncview` closes and reports “NetCDF: HDF error”.

Error may occur if HDF5 cannot find the JPEG-LS decoding filter plugin which is part of the FCI decompression software (see section 2.4.2 for more details on this error case).

Error may also occur if the input netCDF file is corrupted or has been created using a HDF5 filter that has the same filter identifier that the HDF5 JPEG-LS filter of the FCI compression software (see the error messages listed in section 2.4.4).

2.4 Reference manual

2.4.1 Introduction

Once correctly installed on the system, the FCI decompression software decompresses seamlessly the compressed data i.e. without any intervention required by the user. Users can access the compressed data as any other data in netCDF-4 files using any software tool that is dynamically linked to the netCDF-4 C library, for instance ncdump and nccopy utilities or ncview visual browser for netCDF files.

Examples of usage with nccopy and ncview are provided in section 2.5.

2.4.2 Help method

This section describes problems a user may encounter during the use of the FCI decompression software and the solution to these problems.

2.4.2.1 User application makes use of the JAVA implementation of netCDF

Unfortunately, application using the JAVA implementation of netCDF cannot access data compressed with the FCI compression software.

Users are encouraged to use applications linked to the netCDF C library or a programming language that makes use of the netCDF C library. For example there exists netCDF APIs based on the netCDF C version for the following languages: Python, Ruby, Perl, Fortran-2003, MATLAB, IDL, and R (see <http://www.unidata.ucar.edu/software/netcdf/docs/architecture.html>).

2.4.2.2 HDF5 cannot find the HDF5 JPEG-LS filter plugin

This happens if the JPEG-LS filter “H5Zjpegls” of the FCI decompression software is either:

- Not installed on the system;
- Installed in non standard location and the HDF_FILTER_PLUGIN environment variable is not set.

HDF5 returns errors of the following type:

```
HDF5-DIAG: Error detected in HDF5 (1.8.14) thread 0:
...
#004: H5Z.c line 1357 in H5Z_pipeline(): required filter 'HDF5 JPEG-LS
filter' is not registered
  major: Data filters
  minor: Read failed
#005: H5PL.c line 298 in H5PL_load(): search in paths failed
  major: Plugin for dynamically loaded library
  minor: Can't get value
#006: H5PL.c line 402 in H5PL_find(): can't open directory
  major: Plugin for dynamically loaded library
  minor: Can't open directory or file
```

In this case, either install the H5Zjpegls library (the file libH5Zjpegls.so) in the standard HDF5 filter plugin library location “/usr/local/hdf5/lib/plugin” or set the HDF5_PLUGIN_PATH environment variable point to the location where the H5Zjpegls library is installed (path where the file libH5Zjpegls.so can be found).

2.4.2.3 CharLS library cannot be found on the system

This could eventually happen if CharLS library has been moved or deleted of the system after the FCI decompression software installation. In this case, the following error is reported:

```
error while loading shared libraries: libCharLS.so: cannot open shared
object file: No such file or directory
```

CharLS library should be re-installed at its original location, in a standard location (“/usr/lib” or “/usr/local/lib”) or at a location pointed by the LD_LIBRARY_PATH environment variable.

2.4.3 Commands and operations

The FCI L1c decompression software offers the standard netCDF-4 C library interface to the software calling netCDF-4 library routines to decompress and read FCI L1c. It does not extend the netCDF-4 C library interface. It is foreseen that the software calling netCDF-4 library uses the following (non exhaustive) list of standard netCDF-4 C library resources (routines):

- nc_open(): Open an existing netCDF file.
- nc_inq_var_id(): Find the ID of a variable, from the name.
- nc_get_var_short(): Read all the values from a netCDF variable of an open netCDF dataset in short type.
- nc_get_var_uchar(): Read all the values from a netCDF variable of an open netCDF dataset in unsigned char type.
- nc_close(): Close an open netCDF dataset.

The documentation for the standard netCDF-4 C library routines can be found in the netCDF C documentation [RD1].

2.4.4 Error messages

The following messages could be reported to the standard error stream:

2.4.4.1 Messages from the decompression filter: H5Zjpegl

- "Wrong filter direction in the HDF5 JPEG-LS decode filter!"
The decoding filter has not been called with the H5Z_FLAG_REVERSE flag. Data are not decompressed.
- "Fail to allocate memory."
Memory allocation failed for the decompressed buffer. Data are not decompressed.
- "Error during the JPEG-LS decompression of the dataset."
An error occurred during the JPEG-LS decompression of the data. Data are not decompressed.
- "Invalid number of parameters cd_nelmts in the HDF5 JPEG-LS decoding filter."
The input netCDF-4 compressed file is not valid. It may not have been compressed with the FCI compression software. Data are not decompressed.

2.4.4.2 Messages from the façade to CharLS: fcicomp_jpegl

- "Not enough memory allocated for the output of the JPEG-LS decoding process."

The decoding filter has not allocated enough memory to hold the decompressed data. The parameters in the input file may be wrong or the input file is corrupted. Data are not decompressed.

- "The compressed bitstream is not decodable."
CharLS cannot decode the compressed data. The file may be corrupted. Data are not decompressed.
- "Unknown CharLS error code."
CharLS returned an unknown error code. Data are not decompressed.
- "Segmentation fault caught in CharLS!"
A segmentation fault has been caught in CharLS. Data are not decompressed.

2.5 Tutorial

2.5.1 Introduction

This tutorial shows how the FCI decompression software can be used through the following tools:

- The HDF5 h5ls utility. It can be used to examine the contents of a netCDF-4 file. This utility is part of the HDF5 software distribution.
- The netCDF nccopy utility. It copies an input netCDF file to an output netCDF file e.g. with a different compression standard. This utility is part of the netCDF software distribution.
- The ncview visual browser for netCDF format files. It allows displaying the data stored in netCDF files as images.

2.5.2 Getting started

The FCI decompression software decompresses data that have been compressed with the FCI compression software in netCDF-4 files. These data are namely the effective radiance and pixel quality flags in the FCI L1c datasets.

Once correctly installed on the system, the FCI decompression software decompresses seamlessly the compressed data i.e. without any intervention required by the user. Users can access the compressed data as any other data in netCDF-4 files using any software tool that is dynamically linked to the netCDF-4 C library.

For this to work, these software tools should be dynamically linked to the netCDF-4 C library as shown in Figure 2. In this figure, the dynamical links are depicted as arrows. The netCDF-4 C library is dynamically linked to the HDF5 library. Indeed HDF5 is used as the underlying storage layer of netCDF-4 files. Beginning with the version 1.8.11, HDF5 has the capability to dynamically load filters. So if the H5Zjpegls filter of the FCI decompression software is available, it will be dynamically loaded by HDF5 whenever it is necessary, that is to say each time data compressed with the FCI compression filter are being accessed. The H5Zjpegls filter is linked to CharLS library, the optimized implementation of the JPEG-LS, via the fcicomp_jpegls library. This last is an interface to the CharLS library.

HDF5 is able to identify the required filter thanks to information encoded in the netCDF-4 file before the data. For example, the content a netCDF-4 file generated using the FCI compression filter is examined in section 2.5.3.1 using the h5ls utility. It shows information on the filter that has been used for the creation of this file.

The installation process of the FCI decompression software is described in the Software Installation Guide [SIG].

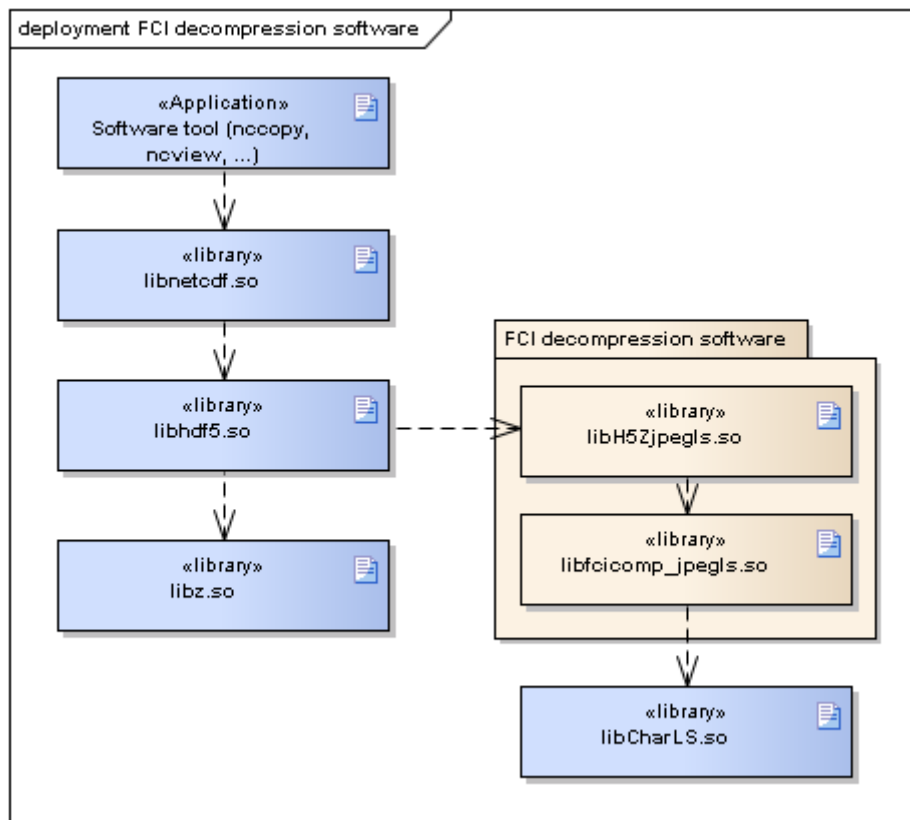


Figure 2: Libraries and links in the FCI decomposition software

2.5.3 Using the software on typical tasks

This section shows how to use the FCI decomposition software through the netCDF application tools to h5ls, nccopy and ncview respectively in order to:

- examine the contents of a netCDF-4 variable and access to its compression ratio;
- copy the compressed netCDF-4 file enabling deflate compression instead of JPEG-LS compression;
- display the effective radiance data.

2.5.3.1 Examining the contents of a netCDF-4 variable and access to its compression ratio

The content of a netCDF-4 variable can be examined using the h5ls utility.

It can be called with the following command to examine the content of the “effective_radiance” variable located in the group “/data/vis_04/measured” in the file “dummy_dataset.nc”.

```
h5ls -v dummy_dataset.nc/data/vis_04/measured/effective_radiance
```

The -v is for verbose option.

This gives the following results:

```
Opened "dummy_dataset.nc" with sec2 driver.
effective_radiance      Dataset {220/220, 2030/2030}
...
Chunks:      {220, 2030} 893200 bytes
```

```
Storage:      893200 logical bytes, 355121 allocated bytes, 251.52%
utilization
Filter-0:     HDF5 JPEG-LS filter-56782 OPT {2, 1, 220, 2030, 16, 1, 0,
0, 0, 0, 0, 0, 0}
Type:        native short
```

The “Storage” field provides information on the data storage. It can be observed that the storage requires 893 200 logical bytes. This is consistent with the data size: one channel of size 220x2030 on 2 bytes (native short type). Nevertheless, thanks to the compression, only 355 121 bytes are needed to store these data. This corresponds to a utilization of 251.52% of the space allocated on disk. Note that the dynamic range of the data is 12 bits and not 2 bytes. The real compression ratio is 1.8825 (12/16 of 251.52%) for this channel.

It can also be observed the HDF5 JPEG-LS filter-56782 filter has been used when writing these data to disk. The number 56782 is the filter identifiers in HDF5. The next numbers are the filter parameters (dimensions of the data and JPEG-LS compression parameters).

2.5.3.2 Copying the compressed netCDF-4 file enabling deflate compression instead of JPEG-LS compression

It is possible to re-compress a FCI L1c dataset using deflate compression instead of JPEG-LS compression using the nccopy utility.

Ncopy utility can be called with the following command to copy the netCDF-4 file “dummy_dataset.nc” to the netCDF-4 file “dummy_dataset_deflate.nc” enabling “shuffle” and “deflate” compression.

```
nccopy -s -d 9 dummy_dataset.nc dummy_dataset_deflate.nc
```

The -s option is to enable the shuffling of variable data bytes before compression or after decompression. The -d 9 option is to enable the deflate compression at its maximum level.

Note: Shuffling can take a long time to run.

2.5.3.3 Displaying the effective radiance data

Data stored into netCDF-4 files can be displayed using the ncview visual browser, even the data compressed in JPEG-LS with the FCI compression software.

Below is an example that shows how to run ncview.

```
ncview dummy_dataset.nc &
```

The effective radiance data of channel VIS 0.9 is displayed below.

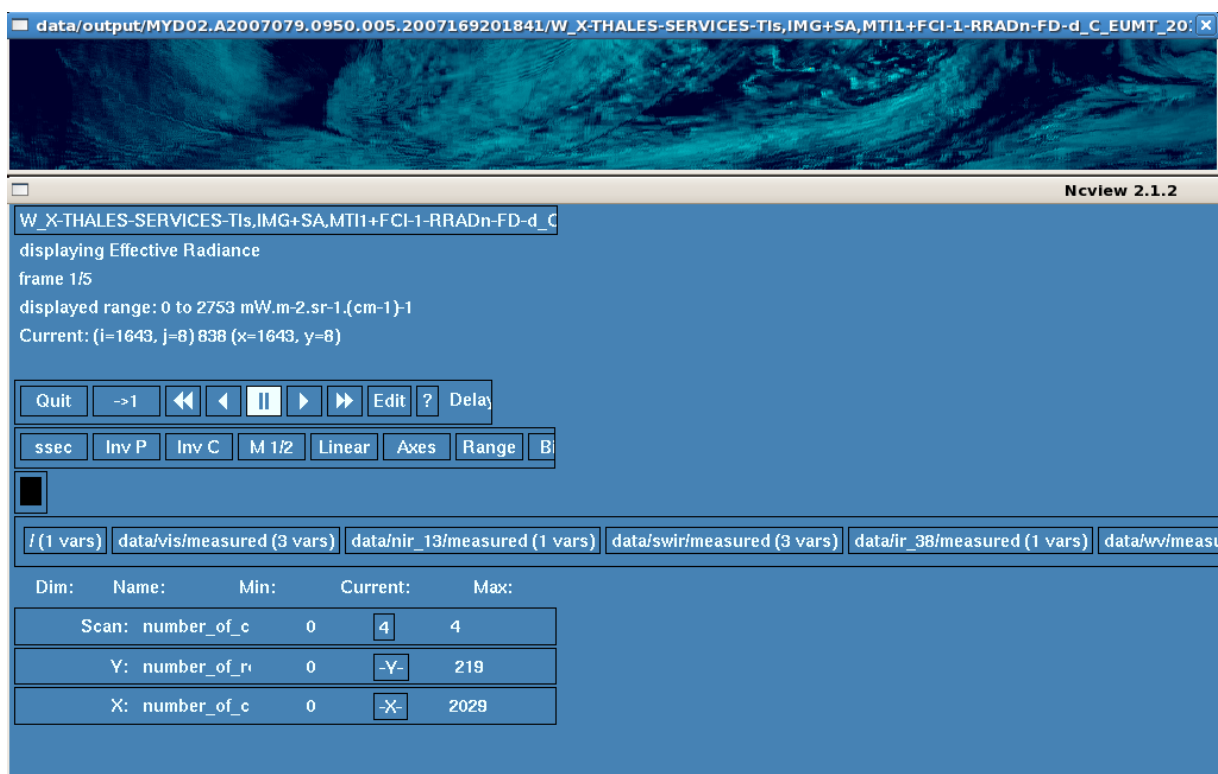


Figure 3: Screenshot of ncview showing the effective radiance data of channel VIS 0.9

END OF DOCUMENT