DW user Documentation

Release 0.1beta

IRA-INAF

Nov 24, 2016

Dish Washer (dw) is a software for RFI (Radio Frequency Interference) identification and mitigation on signals collected by single dish radiotelescope.

Dw is in active development state, be forewarned it may be buggy!

CHAPTER ONE

DESCRIPTION

Dish Washer (DW) is a software for RFI (Radio Frequency Interference) detection and mitigation on signals collected by single dish radiotelescopes.

It is developed at the Institute for Radioastronomy - National Institute for Astrophysics (IRA-INAF), Bologna, Italy

DW is a python package providing tools for detection and flagging of RFI on data collected from single dish radiotelescopes. It provides GUI and command line interface through an interactive python console (i.e. iPython). Currently, together with DW, is shipped *libdw*, a C library aimed to provide efficient implementation of RFI detection algorithms. The library is installed with the python package. In the future the library will eventually distributed independently.

The software is intended to work primarily with data acquired with ...

Dw is in active development state, be forewarned it may be buggy!

CHAPTER

TWO

DATA FORMATS

Currently the supported data format is:

- *fits*: the Flexible Image Transport System

Limited features for *hdf5* (the Hierarchical Data Format 5) support have been included but currenlty they are not working for data flagging.

fits

DW support the *fits* file format with the specifications defined for the Italian radio telescopes, see http://www.ira.inaf.it/Observing/download/MED-MAN-FITS-02.pdf

hdf5

DW does not have a complete support for the *hdf5*.

It was implemented in the fists stages of the developement for testing purpouses for the *roach board* custom output. In the future, if a standard will be defined, the support will be completed.

CHAPTER THREE

USER'S INTERFACES

DW provides basically two types of interface:

- a *Text user's interface*
- a *Graphical user's interface*

All the functionalities of DW are available with both interfaces except interactive data visulization/flagging, available with the *Graphical user's interface* only.

Text user's interface

The text user's interface to DW can be used with any interactive python interpreter.

The following preliminary steps are required to start working with DW in text mode.

Preliminary steps

Importing the module

```
import dw.core.data_def as dw
```

Instatiating a DW class

```
dwc = dw.DWData()
```

Opening and closing data

Opening a data file

```
dwc.open_data(filename, [filetype])
```

filename is a string containing the path to the opening file. *filetype* is a string denoting the fileype. Current available filetypes are:

- *fits*: SRT Nuraghe FITS data format
- *hdf*: DW data format in a HDF5 container
- *hdf_pola*: DW polarimeter format in a HDF5 container (experimental)

Closing a data file

```
dwc.close()
```

Retrieving data and metadata

Read a data table

```
m = dwc.dataset2np_array(i_dataset)
```

i_dataset is the index of the corrensponding dataset

m is an array of 2D numpy arrays which are the data of L and R polarizations

Read a polarimeter data table

```
m = dwc.polaset2np_array(i_dataset)
```

i_dataset is the index of the corrensponding dataset

m is an array of 2D numpy arrays which are the data of the stokes parameters (L, R, Q, U, I, V, Phi, P)

Retrieve data time scale

```
t = dwc.get_time_scale(i_dataset)
```

i_dataset is the index of the corrensponding dataset

t is a 1D numpy array containing the time values of the selected dataset samples

Retrieve data time scale in UT format

```
t = dwc.get_ut(i_dataset)
```

i_dataset is the index of the corrensponding dataset

t is a 1D numpy array containing the time values in UT format of the selected dataset samples

Retrieve data frequencies scale

```
f = dwc.get_freq_scale(i_dataset)
```

i_dataset is the index of the corrensponding dataset

f is a 1D numpy array containing the frequency values of the selected dataset samples

Retrieve data bandwidth

```
b = dwc.get_bandwidth(i_dataset)
```

i_dataset is the index of the corrensponding dataset

b is the dataset bandwidth

Retrieve data ascension

```
a = dwc.get_ascension(i_dataset)
```

i_dataset is the index of the corrensponding dataset

a is a 1D numpy array containing the 'per sample' dataset right ascension

Retrieve data azimuth

```
a = dwc.get_azimuth(i_dataset)
```

i_dataset is the index of the corrensponding dataset

a is a 1D numpy array containing the 'per sample' dataset azimuth

Retrieve data declination

```
d = dwc.get_declination(i_dataset)
```

i_dataset is the index of the corrensponding dataset

d is a 1D numpy array containing the 'per sample' dataset declination

Retrieve data elevation

```
e = dwc.get_elevation(i_dataset)
```

i_dataset is the index of the corrensponding dataset

e is a 1D numpy array containing the 'per sample' dataset elevation

Retrieve pointed source name

```
s = dwc.get_source(i_dataset)
```

i_dataset is the index of the corrensponding dataset

s is a 1D numpy array containing the 'per sample' pointed source name

Retrieve the 'on track' flag

```
t = dwc.get_on_track(i_dataset)
```

i_dataset is the index of the corrensponding dataset

t is a 1D numpy array containing the 'per sample' flag stating whether the source is on track (1 = pointing error < 0.1*HPBW) or not (0 = pointing error > 0.1*HPBW)

Correcting data

Open a correction file

```
dwc.open_correction(file_name)
```

file_name is a correction file generated by BPC (see *Band Pass Correction*)

Toggle corrections on data

```
dwc.toggle_correction()
```

This will switch the correction on (or off) while reading data from datasets. To check if the correction is active see the boolean value of:

```
dwc.correction
```

Flagging data

Create a new flagging set by rectangular area list

```
dwc.new_flagset(i_dataset, flag_areas)
```

i_dataset is the index of the corrensponding dataset

flag_areas is a list of areas to set as flagged (ymin, ymax, xmin, xmax)

Create a new flagging set from a list of arrays

```
dwc.array_flagset(i_dataset, flag_arrays)
```

i_dataset is the index of the corrensponding dataset

flag_arrays is a list of numpy arrays (same shape as data) representing the flagging sets

Retrieve the list of the flagging sets belonging to the selected dataset

```
dwc.get_flagsets(i_dataset)
```

i_dataset is the index of the corrensponding dataset

Read flag data table

```
f = flagset2np_array(i_dataset, k_flagset)
```

i_dataset is the index of the corrensponding dataset *k_flagset* is the key of the flagset in the flagsets dictionary

f is a 2D numpy array of the dimensions of the data which rappresent the matrix of the *k_flagset* of the *i_dataset* dataset

To obtain a dictionary of all the flasgsets of a given dataset

```
d = flagsets2np_array(i_dataset)
```

Merge flagging sets

```
dwc.merge_flagsets(i_dataset, k_flagsets)
```

i_dataset is the index of the corrensponding dataset

k_flagsets is a list of keys of flagsets to merge

Delete flagging sets

```
dwc.del_flagset(i_dataset, k_flagset):
```

i_dataset is the index of the corrensponding dataset

k_flagset is the key of the flagset in the flagsets dictionary

Delete flagging sets intersecating a rectangular area

```
dwc.del_sel_flag(i_dataset, rect_area)
```

i_dataset is the index of the corrensponding dataset

rect_area is a tuple rappresenting a rectangular area: (y_min, y_max, x_min, x_max)

Every flag area intersected by *rect_area* will be removed

Retrieve metadata related to the selected flagging set

```
md = dwc.get_flagset_meta(i_dataset, k_flagset)
```

i_dataset is the index of the corrensponding dataset

k_flagset is the key of the flagset in the flagsets dictionary

md is a dictionary containing the metadata

Copy the flag table to others files

```
dwc.propag_flagtable(filelist)
```

filelist is a list of file paths where the entries of the current flag table will be copied

Automatic flagging

Automatic flagging requires a minimum of two steps to be performed:

- flagging algorithm selection and initialization
- flagging computation

Optionally the user can:

- select algorithm's parameters
- select the algorithm's output (among the algorithm's available selection)

Algorithm selection and initialization

```
dwc.auto_flag_init(i_dataset, alg)
```

i_dataset is the index of the corrensponding dataset

alg is the RFI detection algorithm class.

The available algorthms can be listed using

```
dwc.get_rfi_dect_algorithms()
```

This method returns a dictionary whose keys are the RFI detection algorithm classes. The dictionary contains the name, a short description and a dictionary with the default paremeter's values.

Run the calculation

```
dwc.auto_flag_compute()
```

Flagging matrices are written in the data structure.

Selecting algorithm's parameters

Non default parameters can be selected either by passing a proper dictionary at the initialization

```
dwc.auto_flag_init(i_dataset, alg, **param_dict)
```

or by calling

```
dwc.auto_flag_upd_params(**param_dict)
```

Available parameters are algorithm dependent. Since currently implemented algorithm are for test purpouse, more detailed information about the algorithm's parameters can be found in the developer's documentation.

Selectiong algorithm's output

Each RFI detection algorithm can return more than one flagging matrix.

Available output matrices can be checked without

```
dwc.auto_flag_get_out()
```

The method returns a tuple containing a list with the available option and a dictionary with the current selection.

Output matrices can be selected when invoking the algorithm's computation

```
dwc.auto_flag_compute(out_list)
```

out_list is a list of labels name for the selected output matrices

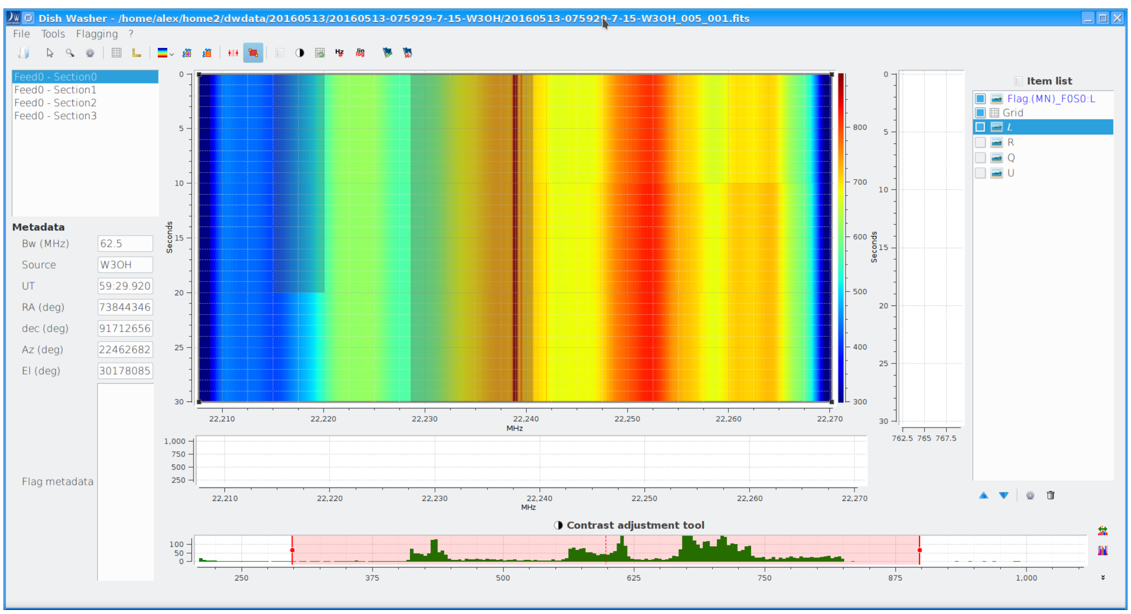
Since currently implemented algorithm are for test purpouse, more detailed information about the output matrices can be found in the developer's documentation.

Graphical user's interface

The graphical users's interface provides access to the DW functionalities available with the text interface and data visualization.

Interface elements

In this Figure an example of the DW GUI is given.



The GUI consists of several panels:

Datasets and metadata

On the left side of the GUI there:

- ***top*: a list of selectable datasets found in the file.** A data file may contain data from a number of feeds, each contaning information from a number of spectral sections. In the above example, the observation has been done with a mono-feed receiver (there is only Feed 0) and contains data from the four different spectral sections delivered by the backend. In this panel one can select the proper combination of feed+spectral section, and the correspondent data are shown in the waterfall plot (see below).

- ***bottom*: a number of metadata characterizing the observation and flagging.** Observational metadata are taken from the headers of the FITS files. Flagging metadata are listed if data already contain a flagging table (see next chapters). The listed metadata are:
  - *Bandwidth:* Is a float number expressing the bandwidth of the dataset, in MHz.
  - *Source:* Name of the observed source.
  - *UT:* UT of the observation.
  - *RA:* Right Ascension (degrees)
  - *Dec:* Declination (degrees)
  - *Az:* Azimut (degrees)
  - *El:* Elevation (degrees)
  - *Flag Metadata:* A list of metadata of the selected flag. These values depends of the method by which the flag was defined.

Item list

On the right of the main window there is an item list that displays data from the combination feed+section that has been selcted in the left panel. Typically, the L, R, Q, U (if present) are listed. Also the flagging regions are listed here, as well as other items like cross sections (see below).

DW differentiates between "visualization" and "action". The *visualization* of an item is regulated by its associated tickmark in the list. To actually *act* (i.e. perform flagging, cross sections etc.) with the available tools on one item, tickmarking is not enough. The item itself has to be highlighted by clicking on it with the mouse, and it becomes highlighted in blue.

The item list is equivalent to a hierarchy of layers in the central panel. If more than one item is tickmarked, the first item in the list is the visible layer.

The flagged regions are shown as grey areas of different transparency depending if they have been already saved in the data or not. Light grey is used for already saved flagging regions.

By default, a Grid item is always present and can be deactivated by un-tickmarking it.

Waterfall plot and cross-sections

At the centre of the GUI the waterfall image of the selected data is shown. The x and y axes of the image represent respectively frequency and time. Default units are respectively MHz and sec. On the sides of the image, there are two panels used to display the cross-section tool results.

Data cross sections are displayed by means of the *Cross section* tool, the *Average cross section*. The Average Cross Section computes and displays the cross section of data inside the selected rectangular region. Non-persistent cross section tool can be activated pressing *Alt* button and moving the mouse pointer over the data.

Data value in a given position can be retrieved by clicking the *Selection* button (the arrow button belo thw Tools menu) and then moving the mouse pointer on the image while keeping the Alt key pressed.

Contrast tool

At the bottom of the GUI there is a panel to control the contrast levels of the item selected from the data list.

As already said, this tool *acts* on the selected item of the data list, so the item must be selected and not only tickmarked.

Menus

File menu

- *Open:* start the open file dialog
- *Close:* close file
- *Quit:* quit DWData

Tools menu

- *Ipython console:* start the Ipython console (currently works only inside spyder for debug purpouse)
- *Band pass correction tool:* Start the BPC tool (see *Band Pass Correction*)
- *Open a correction file:* Load in DW a correction file generated by *Band Pass Correction*
- *Apply/unapply correction:* Apply/remove a correction file generated by *Band Pass Correction*.

Flagging menu

- *Flag:* create a flagging matrix from the selected areas (require area/s is/are selected usign a selecion tool)
- *Deflag intesected:* deflag all previous flags intersecting a rectangle area (require area/s is/are selected usign a selecion tool) **BUG: THIS FEATURE IS NOT WORKING PROPERLY IN CURRENT VERSION**
- *Delete selected:* delete the selected flagging areas
- *Flag widget:* start the flag widget to flag time and/or channels range
- *Propagate flag table to files:* Appy an existing flag table to other data. **NOT IMPLEMENTED YET**
- *Auto RFI detection:* start the automatic RFI detection widget

Toolbar buttons

- *Open file:* start the open file dialog.
- *Selection:* if enabled, by keeping the Alt key pressend data values and coordinates at the cursor position are displayed.
- *Rectangle zoom:* zoom on the selected area
- *Parameters:* open a dialog to set some visualization parameter
- *Grid:* open a dialog to set some grid visualization parameter
- *Axes style:* open a dialog to set some axis visualization parameter
- *Select colormap:* select the colormap for the active data
- *Cross section:* create a cross section item
- *Average cross section:* create an average cross section item. It displays the averaged cross sections of the selected area.
- *Channel range flagging:* manual tool to flag a channel (spectral) range. It creates a flagging item.
- *Area flagging:* manual tool to flag a rectangular. It creates a flagging item.
- *Item list manager:* toggle on/off the item list panel

- *Contrast panel:* toggle on/off the contrast panel
- *Grid:* toggle on/off the data grid
- *Toggle axis:* toggle axis measurement units from (MHz, sec) to (Channel Number -Time Sample).
- *Y scale toggle:* toggle on and off log scale on y axis **NB CHECK THIS **
- *Flag:* save the selected flagging items in the data flagging table. Multiple items can be selected and saved at once by keeping presseld the Shift key.
- *Deflag intesected:* deflag flagged regions intersected by a rectangular area **NOTE IT CURRENTLY ACTS ONLY ON FLAGGED ITEMS NOT ALREADY SAVED IN THE FLAG TABLE**

CHAPTER

FOUR

FLAGGING

Manual flagging

To mannually flag an area of the waterfall plot there are two tools:

- *Channel range flagging*
- *Area flagging*

They will create an element in the data list.

One or more of these elements can be selected and saved as a flagset with the command *Flag* found in the flag toolbar or in the flag menu.

Auto RFI detection

To run the *automatic RFI detection*:

- select *Flagging -> Auto RFI detection*.
- choose the detection algorithm to use and press *OK*
- change the default parameters if necessary, then press *OK*
- choose the flagging matrices to calculate.

Automatic detected RFI matrices are added to the data structure.

To see the available algorithms refer to next Section.

Automatic RFI Detection Algorithms

Currently only one automatic algorithm for RFI detection is implemented.

Simple Threshold Algorithm

Method: sigma-clipping.

Parameters: number of r.m.s. above the median data value.

The user defines the flagging threshold as a multiple of the data r.m.s. To help in this choice, the mean, median and r.m.s. for each data type (L, R, Q, U) are displayed. The user may chose to apply the flagging to one or more data types. In case more data types are to be flagged simultaneously, the proper threshold for each type is computed wit the related r.m.s. The algorithm identifies all the data above the threshold, creating a number of rectangular regions defining the flagged areas. These regions are displayed on the image and listed as a single item in the item list.

CHAPTER

FIVE

BAND PASS CORRECTION

The DW package includes also a tool for the *correction* of the band pass, BPC.

Note that this tool is not for calibration purposes and has not been tested yet.

The uderlying idea is to correct data for the shape of the bandpass, to make it easier the subsequent identification of RFI with dedicated automatic algorithms.

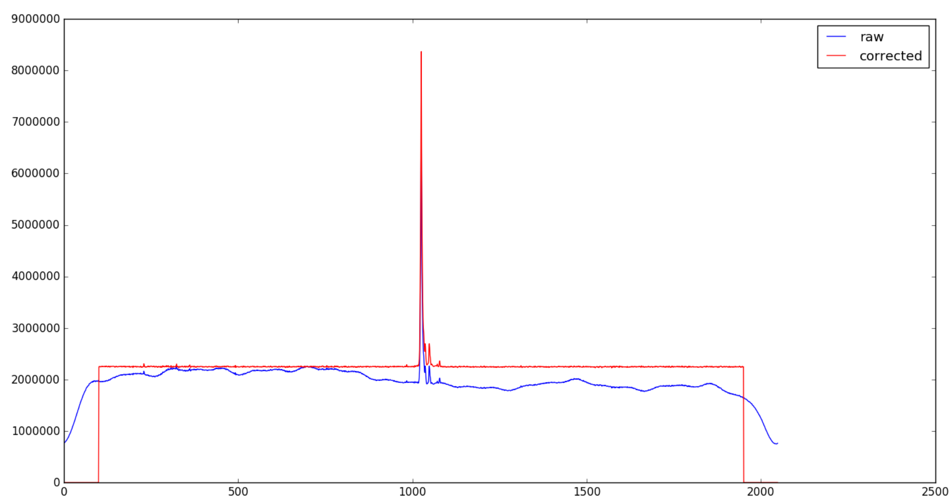
The bandpass shape is computed on raw data that are free from source and RFI signal. Such data must be carefully selected by the user, and are processed to derive a polynomial curve describing the bandpass shape that will be later used to correct the raw data themselves.

All data to be considered to compute a bandpass correction must reside in the same directory. This reflects the way fits data from the same scan are organized, following the scheme in use at the Italian radio telescopes,

The bandpass correction can be subdivided in the following steps:

- Select the useful data by automatically evalutating the median (in the time domain) of every subscan
- Build the bandpass curve by median-combining all the selected data
- Fit the bandpass curve
- Normalize the fit at peak value
- Divide each selected raw data by the normalized curve, create a new corrected data table and append it to the original file

An example of raw and corrected data is shown below.



BPC provide, like DW, two types of interface:

- a *Text user's interface*
- a *Graphical user's interface*

Text user's interface

The text user's interface can be used with any interactive python interpreter.

Peliminary steps

Importing the module

```
import dw.core.data_def as dw
```

This is the same DW module, and it will import also the text mode capabilities of BPC.

Instantiating a BPC class

```
bpc = dw.DWObs()
```

This instantiates a BCP data object.

DWObs is a class with the following attributes:

dir_name is the directory name

files_type is the extension of the files in the direcory

dir_datasets is a list of DW object DWData()

feeds is a list of feeds

sections is a list of sections

polars is a list of polarizations

dssel is the current selection of feed, section and polarization

Opening data

Opening an observation directory

```
bpc.open_dir(dirname, [filestype])
```

dirname is a string containing the path to the directory containing the data to be used for bandpass correction computations.

filestype is a string denoting the fileype of the files in the observation directory. Current available filetypes are:

- *fits*: DW data format in a FITS container

When a directory is opened, all the data files inside that directory are opened as DWData() objetcs and listed in the attiribute *dir_datasets*.

Retrieving data

Getting the median of a datasets

A dataset may be composed by one or more FITS data files.

```
bcp.get_median_data(ldataset, section, pol, lrangei, lranges, lexcluded)
```

ldatasets is a N-list of indices of the considered datasets

section is the index of the working section (starting from 0)

pol is the index of the working polarization (L=0, R=1, Q=2, U=3)

lrangei is a N-list of values of spatial samples at the beginning of each subscan to be used for computation of the median (for OTF scan observations). It assumes that a source, if prpesent, will not fall in the initial and final part of a subscan. A value equal to 0 means to use all the available spatial samples in the subscan.

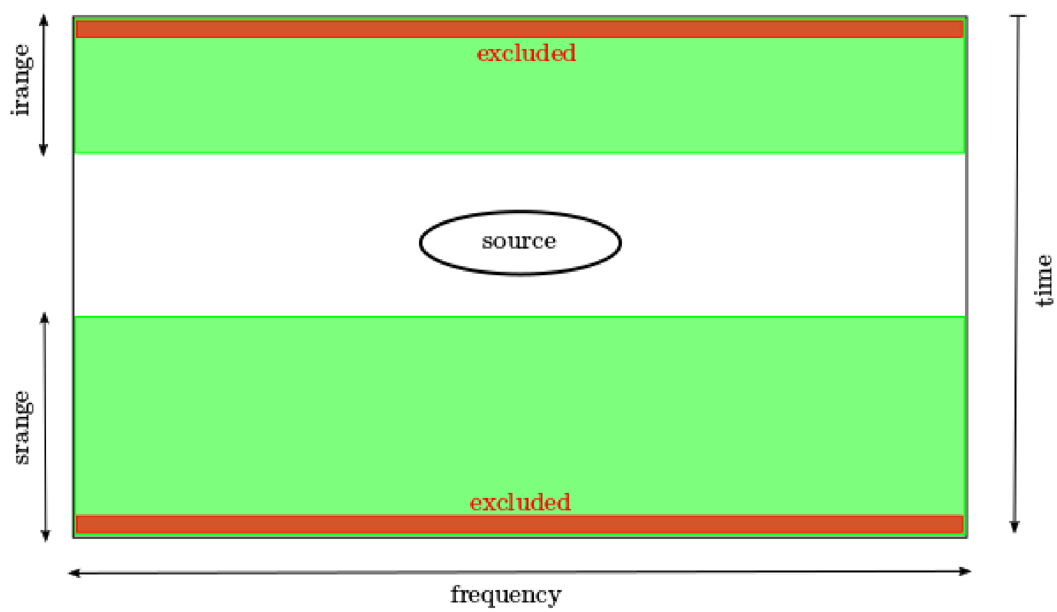
lranges is a N-list of values of spatial samples at the end of the subscan to be used for computation of the median (for OTF scan observations). It assumes that a source, if prpesent, will not fall in the initial and final part of a subscan. A value equal to 0 means to use all the available spatial samples in the subscan.

lexcluded is a N-list of values of spatial samples to be excluded at the beginning and at the end of a subscan.

It returns a list of two np-arrays [x, y].

This function has been designed to compute both the median of the time samples in a single file and the median of several concatenated files.

Once section and polarization are chosen, to build the median three values are needed for each file, provided by the lists *lrangei*, *lranges*, *lexcluded*.



The combination of these three parameters set the sample ranges at the begining and at the end of the subscan (green areas) to be used for median computing. The size of the red areas is described by the *lexcluded* parameter.

For example, on a subscan of *n* time samples if we set:

- *irange* = 100
- *srange* = 200
- *excluded* = 3

to compute the median the first 100 and the last 200 samples, excluding the fisrt and the last 3, will be used. These parameters are useful only in the case of an OTF scan.**NOTE THIS SENTENCE HAS TO BE CLARIFIED**

After the median computation, the values of feed, section and polarization are saved as current working dataset in the *dssel* attribute.**NOTE THIS SENTENCE HAS TO BE CLARIFIED**

Getting a preflag**NOTE THIS DESCRIPTION HAS TO BE CLARIFIED**

This function computes the fit of *data* using the specified fit type.

```
bcp.get_flag_curve(file_name, section)
```

file_name file in which there are the flag data

section is the section in use

It returns a nparraywhose elements are equal to one except for the flagged values, that are set to zero.

This function is useful if RFI is present in every data file and therefore it is not possible to find "clean" data for bandpass correction.

The result can be used as a weight for the fit funcion.

Fitting data

```
bcp.get_fit(fit_type, data, order, smooth, degree, begin, end, weight)
```

fit_type is the fit type. Aviable values are:

- *spline*
- *chebyshev*

data is a list of two nparray [x, y]

order is the spline order, ranging from 1 to 5 (default = 3, cubic spline)

smooth is a positive smoothing factor used to choose the number of knots. The bigger is *smooth*, the smaller is the number of the nodes.The number of knots will be increased until the smoothing condition is satisfied:

$$\sum_i (w(i)(y(i) - spl(x(i))))^2 \leq smooth$$

where *y* are the data, *spl* the spline functions and *w* the weights. The *order* and *smooth* parameters are meaningful only if *fit_type* is "spline" and are ignored otherwise.

degree is the degree of the chebyshev polynomial. This parameter is used only if *fit_type* is "chebyshev".

begin and *end* determine the range (in frequency samples) in which the fit is to be computed.

weight is a optional nparray which is used as weights in the fit computation.

It returns a list of two nparrays [x,y]

The correction curve is normalized at the peak and the points outside the fit range are set to value -1 ("diasabled").

Save an apply to data

The following functions are used to generate and update a bandpass correction file and to apply it to the observation files, generating tables of corrected data.

Generate and update the bandpass correction file

```
bcp.fitfile(file_name, fit)
```

namefile is the path of the new file

fit are the fit data

This function create or, if it already exists, update the file containing the bandpass correction curve and a list of the files of the observation.

Apply the correction to a set of files

```
bcp.applycorr(file_name, filelist)
```

namefile is the path of the correction file *filelist* is a list of path

The function *applycorr* applies the correction found in the *file_name* file to: * all the raw data files in the working directory (typically the scan directory,

> containing a number of subscan FITS files) if filelist is not specified, or
>
> - the files listed in the *filelist* parameter.

A FITS extension table called “CORR DATA TABLE” is created and appended to each raw data file.

This function divides the raw data by the normalized fit curve, setting data to zero where and where the normalized curve is “disabled” (curve value -1).

The CORR DATA TABLE is recognized by DW and can be used for RFI flagging operations.

Usage Example

Here is an example of tipical usage of the bandpass correction tool.

The first step is to instantiate the BPC class and to open the direcory:

```
import dw.core.data_def as dw
bpc = dw.DWObs()
bpc.open_dir("YYYYMMDD-HHMMSS-Project-Suffix/")
```

files and data are now in the attribute *dir_datasets*. For example, if one wants the title of the files in the form “Scan#.Subscan#” it is possible to write:

```
for f in bpc.dir_datasets:
        print f.title
```

To evalutate the median of a dataset some plotting tools are to be imported. For example to compute the median of the third file, feed 0, section 1, polarization R, full spatial sample, one could write:

```
from matplotlib import pyplot as plt
med = bpc.get_median_data([2], 0, 1, 1, [0], [0], [0])
plt.plot(med[0], med[1])
plt.show()
```

Once selected the files to be used (for example 2, 4, 5, 7), a global median can be generated:

```
gmed = bpc.get_median_data([2, 4, 5, 7], 0, 1, 1, [0, 0, 0], [0, 0, 0], [0, 0, 0])
plt.plot(gmed[0], gmed[1])
plt.show()
```

and the fit can be done. For example, chebyshev of 90th degree, from the 100th to the 1900th frequency sample:

```
fit = bcp.get_fit("chebyshev", gmed, 0, 0, 100, 100, 1900)
plt.plot(gmed[0], gmed[1], fit[0], fit[1])
plt.show()
```

Optionally, if all the available raw data are heavily contaminated by RFI and a "clean" dataset for bandpass correction cannot be selected, a correction file computed for a different (clean) dataset may be imported and used:

```
flag = bcp.get_flag_curve("flagged_file.fits", 1)
flag_fit = bcp.get_fit("chebyshev", gmed, 0, 0, 100, 100, 1900, flag)
plt.plot(gmed[0], gmed[1], fit[0], fit[1], flag_fit[0], flag_fit[1])
plt.show()
```

Once the fit is acceptable, we can save the data in a file:

```
bpc.fitfile("correction_filename.fits", fit)
```

or, if flagged **WHAT DOES THIS MEAN?**

```
bpc.fitfile("correction_filename.fits", flag_fit)
```

This step creates the file, if not existing, and fill thhat part of the table relative to the chosen dataset (feed 0, section 1 and polarization R in this example).

Once the bandpass correction file is filled, it is possible to apply it to all the raw data with:

```
bpc.applycorr("correction_filename.fits")
```

or to a selectd list of files:

```
bpc.applycorr("correction_filename.fits", [file1.fits, ..., fileN.fits])
```

Each raw data file will now contain an extension table with the bandpass corrected data.

Graphical user's interface

The Graphical User Interface of the bandpass correction tool provides all the text functionalities plus a graphical rappresentation of data.

The GUI can be launched with the command *bpcgui* or from the "tools" menu in DW.

Interface elements

Menu

File menu

Open: start the "open directory" dialog box.

Quit: quit the Bandpass Correction Tool.

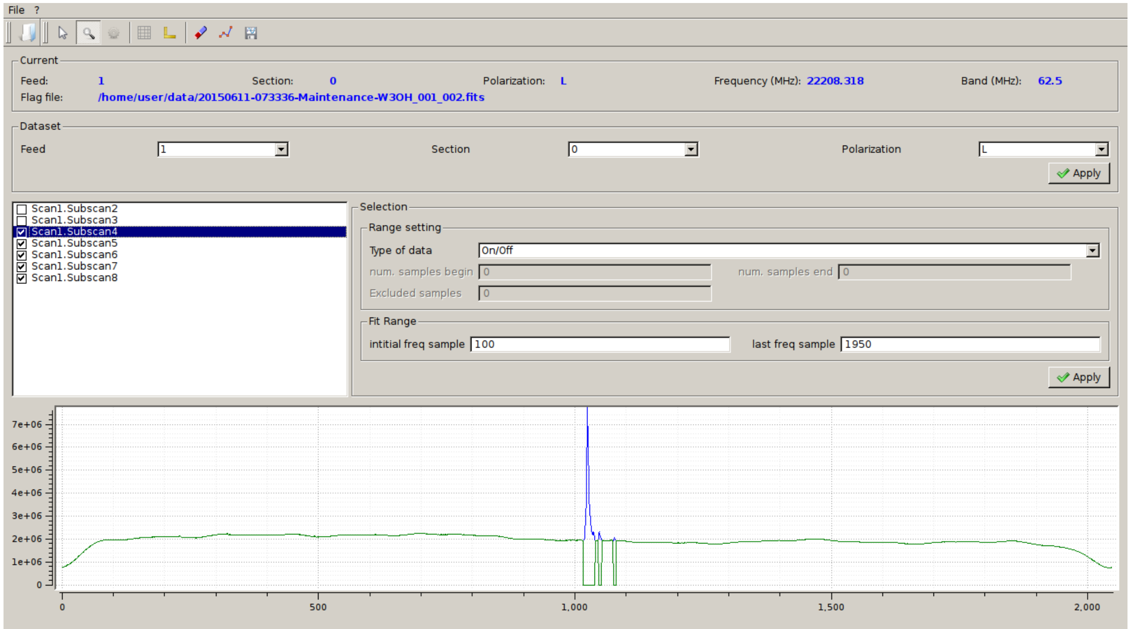
? menu

About: version information

Toolbar buttons

- *Open file:* start the open directory dialog box.
- *Selection:*
- *Rectangle zoom:* zoom on the selected area
- *Parameters:* open a dialog box to set some visualization parameter
- *Grid:* open a dialog box to set some grid visualization parameter
- *Axis style:* open a dialog box to set some axis visualization parameter
- *Get preflag:* open a file preflagged by DW **CLARIFY**
- *Fit:* open the fit window
- *Apply:* apply the correction curve to all, or part of, the data files in the directory

Main window



The main window, as we can see in the picture, is divided in five areas:

Current

A box with information on:

- *feed*
- *section*
- *polarization*
- *frequency*
- *bandwidth*
- *file for the preflag*

Dataset

A selector for the working dataset:

- *feed*
- *section*
- *polarization*

The *apply* button set the selected dataset. It will set the *dssel* parameter described in *Text user's interface*.

List

A list of files in the observation directory, selectable by means of checkboxes.

Here it is possible to select the files to be used for the correction curve (**NOTE: also its application?**).

Selection

The selection box is divided in two subboxes:

- *Range settings*
- *Fit range*

The *range setting* can be used to set the type of observation and to set a range of spatial samples to be used for the median computation:

- *type of data:* the type of observation. Values are: "On/Off", "Cross Scan" and "Map"
- *select region inf:* set a number of spatial samples at the beginning of the data to be used for the median computation. A value equal to 0 means to use all available samples.
- *select region sup:* set a number of samples at the end of the data to be used for the median computation. A value equal to 0 means to use all available samples.
- *excluded region:* set a number of samples at the beginning and at the end of the data to be excluded from the computation (useful to exclude ramps).

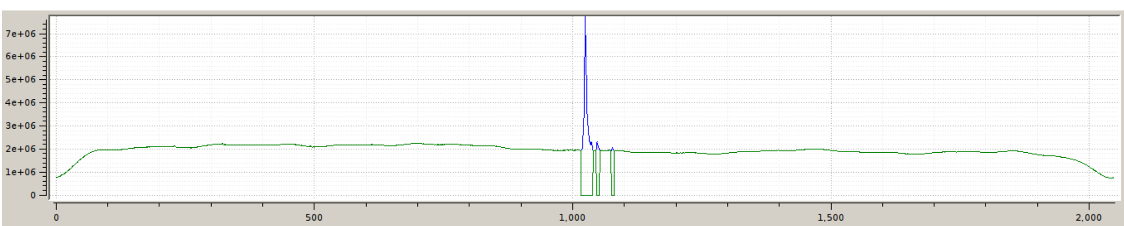
The *select region inf* and *select region sup* are active only if *type of datas* is "Cross scan" or "Map".

The *fit range* parameter sets a range for the computation of the fit. The fit is computed in the region included within *select fit region inf* and *select fit region sup*.

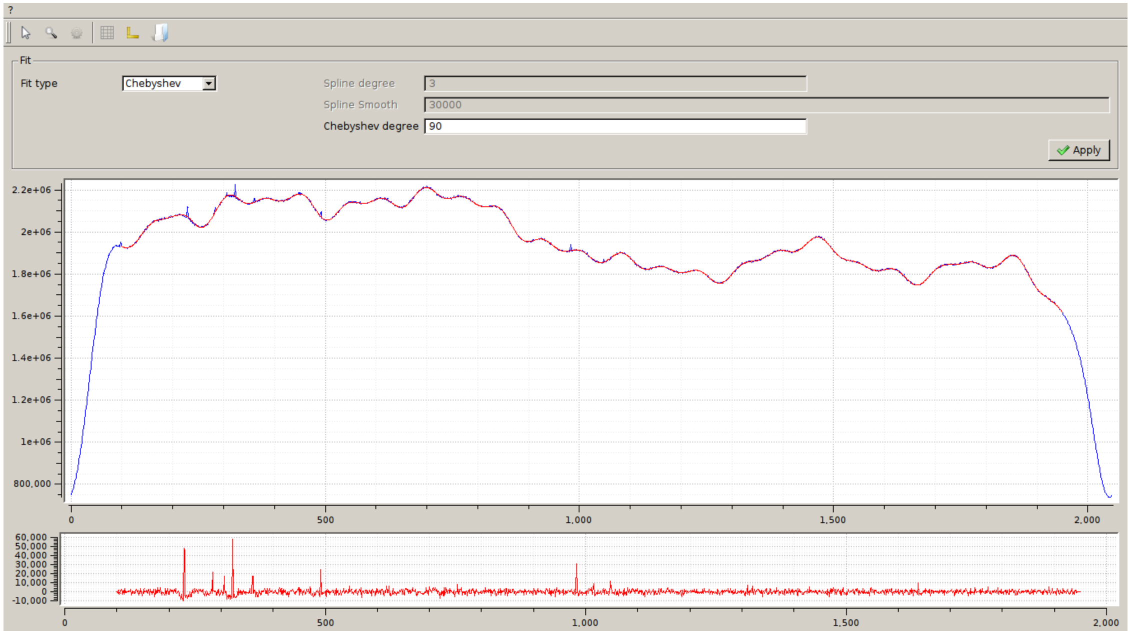
The *apply* button saves these settings and updates the plot.

Plot area

The area in which the median plot (blue) and median "cut" by preflagging, if applied (green) **[???]** are shown.



Fit Window



In the toolbar of the Fit Window the folowing buttons are available:

- *Selection:*
- *Rectangle zoom:* zoom on the selected area
- *Parameters:* open a dialog box to set some visualization parameter
- *Grid:* open a dialog box to set some grid visualization parameter
- *Axis style:* open a dialog box to set some axis visualization parameter
- *Save fit:* save the current fit to the "fit_data_correction.fits" file in the root direcory.

The fit window is divided in two areas:

- *fit settings*
- *view area*

Fit settings

These are:

Fit type is a selector for the type of the fit. Current supported types are: * *Spline* * *Chebyshev*

Spline degree and *Spline smooth* are free parameters for the spline method. The meaning of these parameters is explained in the *Text user's interface* section.

Chebyshev degree is the degree of the Chebyshev polynomes.

The button *apply* starts the computation and visualizes the result in the plot area.

Plot area

It is the area for visualizing the data (blue) and the fit (red).

Usage example

Operations needed to build the the bandpass correction file and to apply it to the raw data are the same of the text user interface case.

Once opened the directory of the observation, files browsing is possible thanks to the list widget. Also, selecting feeds, spectral sections and polarizations is possible using the dataset box.

In the case of OTF observing modes ("Cross scan" or "Map") the ranges of spatial samples to use are to be set, as explained in *Text user's interface*.

Tthanks to the checkboxes on the list widget, only the files whose median does not present excessive irregularities can be selected.

The fit window can be opened by means of the toolbar button. The window will show the median of all the selected files.

Some tests to obtain the best fit possibile can be done, and result can be finally saved by means of the toolbar button. Fit data will be saved in the file "fit_data_correction.fits" file.

Once closed the fit window, one can perform the same operation on some or all the feed/section/polarization combinations. Finally, the bandpass correction can be applied to all the raw data files and for each of them the FITS extension table containing corrected data is written.

Correction file

To be done.

CHAPTER

SIX

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program–to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to “keep intact all notices”.

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation’s users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor’s “contributor version”.

A contributor’s “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, “control” includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor’s essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a “patent license” is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To “grant” such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. “Knowingly relying” means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient’s use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others’ Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

> <one line to give the program's name and a brief idea of what it does.> Copyright (C) <year> <name of author>
>
> This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.
>
> This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
>
> You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

> <program> Copyright (C) <year> <name of author> This program comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

CHAPTER

SEVEN

SEARCH:

- search