

Real-time Skew-T Plotting

See the *Chart of Near Realtime GTS BUFR Sounding Data Processing and Plot Generation for OTREC Field Project at the end of this document.*

Directory Structure (under /net/work/Projects/Project-name, where Project-name is OTREC, HIGHWAY, etc.).

real-time-plots (contains entire processing for the real-time skew-t plots)

software (contains all software required)

 src (needs to exist, but has nothing in it)

 data_processing

 logs

 output_ascii_data

 output_esc

 output_plots

 output_preproc_data

 raw_gts_bufr_data

 snapshots

This directory structure is already set up when checking out this software from svn:
(dmg/tools/upper_air/skewt_plot_realtime).

**Also needed for processing the bfr files is the GTS_BUFR software available in svn at
dmg/conversions/upper_air/GTS_BUFR/tags/SWEX_2020_realtime. This is the latest
version of the software as of 3/13/2025, which was run for SWEX in 2022. The files
needed are as follows:**

GTSBUFR_Radiosonde_Converter_RealTime.pl

preprocess_GTS_BUFR.pl

These files are placed in the real-time-plots/software directory.

There are two important files under the data_processing directory:

DO_NOT_DELETE_last_run_time

DO_NOT_DELETE_running

The last_run_time file keeps track of when the process last successfully processed bfr files.

This is used to decide which bfr files have not yet been processed. The running file is used to denote that the process is currently running. When the next process starts, if the running file exists, then the new process stops. Two processes should not be run at the same time. If one of the programs crashes, then the running file may have to be deleted. (Make sure the Real-time-plots.pl program is not running before deleting the file). The process runs on eol-hurricane as the joss user (sudo su - joss to become the joss user). The process runs from the crontab with an example from OTREC shown below:

```
0,10,20,30,40,50 * * * * /net/network/Projects/OTREC/real-time-plots/software/Real-time-plots.pl
```

The crontab was changed for ACCLIP (test only) to run as a specific group with the following command:

```
#0,10,20,30,40,50 * * * * /bin/sg ctm-dmg  
/net/network/Projects/ACCLIP/real-time-plots/software/Real-time-plots.pl
```

This command was tested and determined that it works, but it has not yet been used in a production environment.

For each project, the software will need to be changed to update the directory names. This includes Real-time-plots.pl, build.xml, plot_sounding.py, and convertESCTosharppy.pl.

The raw_gts_bufr_data, output_preproc_data, output_ascii_data, and output_esc directories are used by the GTS_BUFR conversion software. This software can be found on svn at dmg/conversions/upper_air/GTS_BUFR. The logs, output_plots, and snapshots directories are used by the skew-t plotting software. This software can be found on svn at dmg/tools/upper_air/skewt_plot_realtime.

The process:

- 1) The bfr files are ftp-ed to a directory. Greg's process may rename the bfr files if they don't have the correct naming convention.
- 2) The real-time skew-t software (Real-time-plots.pl) reads the ftp directory to determine which bfr files need to be processed. The files, to be processed, are copied to the raw_gts_bufr_data directory.
- 3) The GTS_BUFR conversion software is run to convert the bfr files to ESC format.
- 4) The ESC files are sorted by time. The ESC files are in the output_esc directory.
- 5) The convertESCTosharppy.pl program is run to convert the ESC files to a format that the sharppy program expects. These cls.sharppy data files are also in the output_esc directory.
- 6) The call_plot_sounding.py program is run to set up the python environment and then calls the plot_sounding.py program to create the skew-t plot, which is put in the output_plots directory.
- 7) After all the data files have been processed, all of the plots in the output_plots directory are copied to the field catalog incoming directory.
- 8) The process tars and gzips all the directories in the data_processing directory (except for snapshots) and places the resulting file in the snapshots directory. This is done to make it easier to debug issues that happen with the data files, either rerunning or determining what errors happened. There is a raw-data-cross-reference.txt file which is in the snapshots directory. This file lists the bfr files that were processed with each snapshot

tar-gzip file. This is used to determine which snapshot file has the bfr file that did not process. Sometimes the bfr file can be touched in the ftp area, and then it will be rerun when this entire process runs again. That may solve some of the problems, especially if the file was not complete when it was copied to the raw_gts_bufr_data directory to be processed.

Note: The following software programs need to be added to the software directory from the GTS_BUFR conversion software (See above for the svn location):

GTSBUFR_Radiosonde_Converter_RealTime.pl
preprocess_GTS_BUFR.pl

The preprocess... program uses the bufr_dump software located in /opt/local/bufr/bin. (See below for installing this software).

The file snapshots/raw-data-cross-reference.txt needs to exist and is empty at the beginning. When the program runs, it adds the information to this file for debugging purposes as explained above. The DO_NOT_DELETE_last_run_time file has the information from a different project, but can be used as is for the next project.

Installing ecCodes software

The ecCodes software supplies the bufr_dump software necessary for decoding the bufr files. The latest version of the ecCodes software was downloaded from [EcCodes Releases](#). Click the “Download the latest tarball” button. This downloaded eccodes-2.39.0-Source.tar.gz. Copy this tarball to the /opt/local/bufr directory and unpack it. Then do the following in the /opt/local/bufr directory:

```
mkdir build  
cd build  
cmake -DCMAKE_INSTALL_PREFIX=/opt/local/bufr ..../eccodes-2.39.0-Source  
make  
ctest  
make install
```

Chart of Near Realtime GTS BUFR Sounding Data Processing and Plot Generation for OTREC Field Project

