Cirrus CLI

User's Manual

Developed by

http://www.clusterlogic.co.il

Introduction

The Cirrus project unifies several meteorological database queries and manipulations under one simple CLI (Command-Line Interface). It automates the creation of multiple data files and maps from the databases with various parameters.

In order to allow for Cirrus to do it's job, the original scripts (GrADS & MATLAB) have been modified or translated into PERL. However, the modified scripts can still be run manually, if needed. All of the MATLAB scripts have been translated to PERL, but the originals are also provided and can also be run manually. In addition, the MATLAB cloudsat_to_era conversion scripts can also be run through PERL instead of using the PERL versions in case there's a problem with the results.

Cirrus uses a configuration file, which can be selected using the CLI. Thus, several users can run Cirrus, each with his own configurations (data files, result paths, etc.)

Table of Contents

1. The Cirrus CLI File Structure

The following Cirrus CLI files and directories are located under the main Cirrus CLI path:

cirrus	The main Cirrus CLI script.
gsindent	A utility that indents GrADS files. **This is a helper utility which is NOT used by the Cirrus CLI**.
hdfSDreads	A statically compiled utility that reads SD datasets from hdf files
scripts/	This path contains all the modifies HIRS, ERA-Interim and CloudSat scripts (GrADS and MATLAB).
CirrusModules/	Cirrus sub modules for each one of the databases and utility modules.

The modules MUST be located in the CirrusModules subdirectory. However, using the configuration file (see description in Chapter 2), the located of the database scripts can be changes.

The hdfSDreads utility uses the libmfhdf.so.4 library to read hdf files. This utility is statically linked so this library doesn't have to be installed. It's a 64-bit executable. If needed, both 32-bit version or dynamically linked version can be supplied.

2. The Configuration File

The configuration file used by Cirrus is a simple PERL hash dump. Is defines the locations of the various external utilities used by Cirrus, locations of Cirrus sub-scripts, result paths, etc. A basic configuration file looks like this:

```
$configs = {
            # GRADS config
            'gradsPath'     => "/Path/To/Cirrus/src/grads/grads",
            'gradsData'     => "/Path/To/Cirrus/src/grads",
            'gradsDirs'     => "/Path/To/Cirrus/src/gradslib",

            # cirrus modified scritps
            'cirrusScripts' => "/Path/To/Cirrus/cirrus/scripts",

            # HIRS config
            'hirs_data'     => "/Path/To/Cirrus/src/hirs",

            # ERA config
            'era_data'      => "/Path/To/Cirrus/src/era",
            'era_netcdf'    => "/Path/To/Cirrus/db/ERA-Interim/NETCDF_file",

            # cloudsat config
            'cloudsat_db'   => "/Path/To/Cirrus/db/cloudsat",

            # Results Path
            'outPath'       => "/Path/To/Cirrus/results",
           };
```

Lines beginning with '#' are ignored. The following table described the variables and their meaning:

Name	Description
gradsPath	Path to the GrADS executable.
gradsData	Path to the GrADS data files.
gradsDirs	Extra paths to pass on to GrADS where additional scripts and functions can be found (e.g. cbar.gs, colors_scale_of_clouds.gs, etc.)
cirrusScripts	Path to external scripts. These will usual be found in a directory called 'scripts' under the main Cirrus CLI path.
hirs_data	Path to the HIRS ctl and bin files
era_data	Path to the ERA-Interim sources (for the various ctl

	and bin files)
era_netcdf	Path to the ERA-Interim netcdf files
cloudsat_db	Path to the location of the CloudSat zipped hdf files
outPath	Where the results should be located. This path will automatically be created, as well as subdirectories categorized by the database and script used.
matlab	Path to the MATLAB executable. This will only be used when the user whishes to run the modified CloudSat MATLAB scripts instead of the translated PERL versions.
cloudsat_cloud	Path CloudSat cloud zipped hdf files. Overrides cloudsat_db.
cloudsat_repo	Path to cloudsat ctl and bin files. If not given, the files will be saved in 'outPath/{db}/{script}/tmp'.
cloudsat_cloud_extract	Path into which the zipped files will be extracted. If not given, 'outPath/{db}/{script}/zip' will be used.
cloudsat_ice	Path CloudSat ice zipped hdf files. Overrides cloudsat_db.
cloudsat_ice_extract	Path into which the zipped files will be extracted. If not given, 'outPath/{db}/{script}/zip' will be used.
cloudsat_temperature	Path CloudSat cloud zipped hdf files. Overrides cloudsat_db.
cloudsat_temperature_extract	Path into which the zipped files will be extracted. If not given, 'outPath/zip' will be used.

The configurations used by Cirrus are always displayed when running the CLI, which helps to verify they are correct. If a configuration file is not given, Cirrus uses some built-in values, that assume paths are located relative to the CLI's path.

3. The Cirrus CLI

The Cirrus CLI's flags are given in two parts. First the user can select flags for the the main CLI, followed by '--' and then flags for the specific database module. At any point, if no flags are given (before or after the '--') a usage message will be printed explaining the various flags for the appropriate section:

```
Usage: ./cirrus [OPTIONS] -- [DB OPTIONS]

Options:
      --conf {file}    Specify a custom configurations file
      --db   {db}      The db to use (cloudsat, era, hirs)
      -v               Verbose mode
      -d               Debug Mode (also implies verbose)
      --help           This message

For each db, you can provide the '--help' option as a db option to
get a db specific help message:

  ./cirrus --db hirs -- --help
```

The '--db' flag is mandatory. The verbose mode displays extra information to help understand what Cirrus is doing. This is especially helpful in multi-part runs where several scripts are used in succession.

The debug mode is used to debug errors in an execution chain. When used, Cirrus will print all the commands it runs and their results, including ALL of the output produced by external utilities and programs.

Upon execution, Cirrus will attempt to load the requested db module and pass it's flags to it. Loading a module can fail for one of two reasons: either the module doesn't exist, or the flags chosen were incorrect or conflicting. In any case, an appropriate error message will be displayed.

4. The HIRS module

The HIRS module can be selected by giving the '**--db hirs**' flag to the main CLI:

```
> ./cirrus --db hirs --
Can't locate config file, using defaults.
Loading module for db 'hirs'...
Parameters for HIRS database:

      --stats           do monthly stats instead of single month
  -y, --years           years do use (e.g. 1979-1980,1999)
  -m, --months          months to use (e.g. 1-4,8)
  -p, --pressure        pressure levels:
                        950,900,800,700,600,500,440,300,200
  -c, --coords          coordinates as s,n,w,e or lat,lon for
                        vertical profile and grid point
  -v, --vertical        create a vertical profile (nothing else)
  -g, --grid            create a grid point (nothing else)
  -u, --update          update the appropriate coulddat.ctl to
                        correctly find it's coulddat.dat file
                        (add '^' to the filename)
      --help            This message
```

Given the '--stats' flag, Cirrus will use the '**stats_monthly-1979-2001-horiz-vert.gs**' script, otherwise it will use the '**single-month-horiz-vert.gs**' script. When using the '--stats' option, Cirrus will look for the '**cloudave**' files instead of '**clouddat**'.

By default, GrADS looks for the 'dat' file in the current working directory. To overcome this and tell GrADS to look for the 'dat' file in the same place where the 'ctl' file is located, a '^' has to be added to the beginning of the 'dat' filename in the 'ctl' file. Cirrus checks for this and will warn the user when this is not the case. Cirrus can also correct this and add the '^' by using the '-u' flag.

Here is an example of a simple run:

```
> ./cirrus --db hirs --conf tal.conf -- -y 1980..1985 -m 2..4,6 -p 300,200 -c
30,40,30,40
configs:
  cirrusScripts  = /Path/To/Cirrus/cirrus/scripts
  cloudsat_db    = /Path/To/Cirrus/db/cloudsat
  era_data       = /Path/To/Cirrus/src/era
  era_netcdf     = /Path/To/Cirrus/db/ERA-Interim/NETCDF_file
  gradsData      = /Path/To/Cirrus/src/grads
  gradsDirs      = /Path/To/Cirrus/src/gradslib
  gradsPath      = /Path/To/Cirrus/src/grads/grads
```

```
  hirs_data      = /Path/To/Cirrus/src/hirs
  matlab         = /opt/matlab/bin/matlab
  outPath        = /Path/To/Cirrus/results
  path           = /Path/To/Cirrus/cirrus

Loading module for db 'hirs'...
Options:
Years: 1980,1981,1982,1983,1984,1985
Months: 2,3,4,6
Pressure Levels: 300,200
Coordinates:
    South = 30
    North = 40
    East  = 30
    West  = 40
Using Script: single-month-horiz-vert

Running sequence...
Running [year = 1980 , month = 2 , level = 300]
Running [year = 1980 , month = 2 , level = 200]
Running [year = 1980 , month = 3 , level = 300]
Running [year = 1980 , month = 3 , level = 200]
Running [year = 1980 , month = 4 , level = 300]
Running [year = 1980 , month = 4 , level = 200]
Running [year = 1980 , month = 6 , level = 300]
Running [year = 1980 , month = 6 , level = 200]
Running [year = 1981 , month = 2 , level = 300]
Running [year = 1981 , month = 2 , level = 200]
Running [year = 1981 , month = 3 , level = 300]
Running [year = 1981 , month = 3 , level = 200]
Running [year = 1981 , month = 4 , level = 300]
Running [year = 1981 , month = 4 , level = 200]
Running [year = 1981 , month = 6 , level = 300]
Running [year = 1981 , month = 6 , level = 200]
...
Running [year = 1985 , month = 3 , level = 300]
Running [year = 1985 , month = 3 , level = 200]
Running [year = 1985 , month = 4 , level = 300]
Running [year = 1985 , month = 4 , level = 200]
Running [year = 1985 , month = 6 , level = 300]
Running [year = 1985 , month = 6 , level = 200]
Sequence completed successfully!
```

5. The ERA Module

The ERA-Interim module can be selected by giving the '**--db era**' flag to the main CLI:

```
> ./cirrus --db era --
Loading module for db 'era'...
Parameters for ERA-Interim database:

  -s, --stats           do monthly stats instead of single month
                        (ignores -y)
  -t, --type            type of data to use (cloud, ice, liquid,
                        temperature, humidity)
  -y, --years           years do use (e.g. 1979-1980,1999) (ignored
                        when using 'stats')
  -Y, --single-year     use data from single year files (for
                        vertical profile)
  -m, --months          months to use (e.g. 1-4,8)
  -d, --days            days to use (e.g. 1-15,20-23,27)
  -p, --pressure        pressure levels to use
  -P, --percentile      select a percentile (implies stats)
  -h, --hour            hours to calculate mean for (0,6,12,18)
                        omit for all means
  -c, --coords          coordinates as s,n,w,e or lat,lon for
                        vertical profile and grid point
  -v, --vertical        create a vertical profile (nothing else)
  -g, --grid            create a grid point (nothing else)
  -M, --maps            create maps from the percentile data
                        (implies --percentile)
  -n, --min             find average minimum in output files (only
                        with 'stats')
  -u, --update          update the appropriate coulddat.ctl to
                        correctly find it's coulddat.dat file (add
                        '^' to the filename)
      --help            This message
```

This module can use several scripts according to the parameters it's given:

--stats and --percentile	era_stats_monthly
--stats	era_stats_monthly_no_percentile
--vertical and --single-year	era_data_vertical_years
--vertical	era_data_vertical
otherwise	era_data

As a result, '--maps' implies '--percentile' and '--stats'. '--min' also implies '--stats'. Also, '--vertical' can not be combined with '--stats' or '--grid'. Appropriate error messages will be displayed in case these conditions are violated.

The first step in an ERA cycle is to run the GrADS script specified by the selected options. If the '--maps' option is selected, the second step will be to use the output of the statistics (with percentile) script to create new 'ctl' and 'dat' files for the map creation GrADS script. Once this is done, Cirrus will run the map creation scripts to produce the percentile maps.

Given the '--min' option, the final step will be to go over the text files produced by the statistics scripts, find the minimum average value and created a '.min.txt' file containing all the lines from the original file with that minimum value. This is done for each one of the resulting '.txt' files.

Here is an example of an ERA run:

```
> ./cirrus --db era --conf tal.conf -- -s -t cloud -m 2-4 -c 30,40,30,40 -M -u
-p 400 -P 0.0426
configs:
  cirrusScripts  = /path/to/Cirrus/cirrus/scripts
  cloudsat_db    = /path/to/Cirrus/db/cloudsat
  cloudsat_repo  = /path/to/Cirrus/repo
  era_data       = /path/to/Cirrus/src/era
  era_netcdf     = /path/to/Cirrus/db/ERA-Interim/NETCDF_file
  gradsData      = /path/to/Cirrus/src/grads
  gradsDirs      = /path/to/Cirrus/src/gradslib
  gradsPath      = /path/to/Cirrus/src/grads/grads
  hirs_data      = /path/to/Cirrus/src/hirs
  matlab         = /opt/matlab/bin/matlab
  outPath        = /path/to/Cirrus/results
  path           = /path/to/Cirrus/cirrus

Loading module for db 'era'...
Options:
Months: 2,3,4
Pressure Levels: 400
Calculating 0.0426% percentile
Coordinates:
    South = 30
    North = 40
    East  = 30
    West  = 40
Using script: era_stats_monthly

Running sequence...
Running [month = 2 , level = 400]
Running [month = 3 , level = 400]
Running [month = 4 , level = 400]
Creating map for 'infogrid_02_daily_400mb_30_N_40.5_N_30_E_40.5_E'...
```

```
Creating map for 'infogrid_03_daily_400mb_30_N_40.5_N_30_E_40.5_E'...
Creating map for 'infogrid_04_daily_400mb_30_N_40.5_N_30_E_40.5_E'...
Sequence completed successfully!
```

6. The CloudSat Module

The CloudSat module can be selected by giving the '**--db cloudsat**' flag to the main CLI:

```
> ./cirrus --db cloudsat --
Loading module for db 'cloudsat'...
Parameters for CloudSat database:

  -s, --stats          do monthly stats instead of single month
                       (ignores -y)
  -S, --start          start date as 'dd-mm-yyyy'
  -E, --end            end date as 'dd-mm-yyyy'
  -t, --type           type of data to use (cloud, ice, ice_mgm3,
                       temperature)
  -p, --pressure       pressure levels to use
  -d, --diff           cloud differences in absolute value (0-
                       100%) you can specify several:
                       10,30,56.7,90 (only for type 'cloud')
  -c, --coords         coordinates as s,n,w,e or lat,lon for
                       vertical profile and grid point
  -o, --minobs         minimum number of observations required
  -M, --maps           create maps from the percentile data
                       (implies --percentile)
  -n, --min            find average minimum in output files (only
                       with 'stats')
  -z, --unzip          unzip the data files (only if not already
                       extracted)
  -Z, --force-unzip    if case of the perl cloudsat_to_era, force
                       extraction (implies -z)
  -f, --force-ctl      force the creation of ctl and bin files
                       (default is to create only if they don't
                       exist)
      --help           This message
```

The first step in a CloudSat run it to convert CloudSat HDF data into GrADS data so it can be compared to the ERA-Interim data. This requires the zipped HDF files. By default, it is assumed that the zipped files have already been unzipped into the 'extract' path (see the configurations file section). However, the '**--unzip**' option will tell Cirrus to check whether the unzipped HDF files exist or not, and if not, Cirrus will attempt to unzip the corresponding zip files. The '**--force-unzip**' tells Cirrus to skip this checkup and always try to unzip the HDF files. In any case, Cirrus will only select those zip/HDF files that match the given parameters and will skip the rest regardless of the selected options.

The conversion creates 'ctl' and 'bin' files, which will be the input for GrADS in the second step of the run. Since the conversion process itself can take a significant amount of time to complete, the default behavior is to check whether the 'ctl' and 'bin' files already exist, and if so, the conversion is skipped. The '--force-ctl' option bypasses this checkup and forces Cirrus to perform the conversion.

The second step is to run the GrADS script specified by the selected options. The script used will be named 'cloudsat_{type}.gs', or 'cloudsat_{type}_stats.gs' if the '--stats' option is used.

If the '--maps' option is selected, the third step will be to use the output of the statistics scripts to create new 'ctl' and 'dat' files for the map creation GrADS scripts. Once this is done, Cirrus will run the map creation scripts to produce the requested maps.

Given the '--min' option, the final step will be to go over the text files produced by the statistics scripts, find the minimum average value and created a '.min.txt' file containing all the lines from the original file with that minimum value. This is done for each one of the resulting '.txt' files.

The PERL version of the conversion scripts was designed to be compatible with the MATLAB version. However, it seems that PERL uses higher precision in it's calculations and thus, the resulting data is more accurate than that of MATLAB. This can lead to unforeseen consequences. For this reason, two additional 'undocumented' flags exist, that can be used in this module. These flags are meant only for debugging. By default, Cirrus uses it's built-in PERL versions. However, if the need arises, the original MATLAB version can be used via the '--matlab' flag. For debugging purposes, the '--perl' flag can be added to force Cirrus to run both the MATLAB version and the PERL version. Note, that the PERL output files will overwrite those of the MATLAB version. Together with the debug mode, these flags can help compare the work of the PERL version to that of the MATLAB version in case of problematic or erroneous output as described above.

Here is an example of a CloudSat run:

```
> ./cirrus --db cloudsat --conf tal.conf -- -s -t cloud -S 20-06-2006 -E 23-06-
2006 -p 200,250,400 -d 70 -c 20,50,20,50 -o 1 -z -s -n
configs:
  cirrusScripts  = /path/to/Cirrus/cirrus/scripts
  cloudsat_db    = /path/to/Cirrus/db/cloudsat
  cloudsat_repo  = /path/to/Cirrus/repo
  era_data       = /path/to/Cirrus/src/era
  era_netcdf     = /path/to/Cirrus/db/ERA-Interim/NETCDF_file
```

```
  gradsData      = /path/to/Cirrus/src/grads
  gradsDirs      = /path/to/Cirrus/src/gradslib
  gradsPath      = /path/to/Cirrus/src/grads/grads
  hirs_data      = /path/to/Cirrus/src/hirs
  matlab         = /opt/matlab/bin/matlab
  outPath        = /path/to/Cirrus/results
  path           = /path/to/Cirrus/cirrus

Loading module for db 'cloudsat'...
Options:
Pressure Levels: 200,250,400
Diff Percentages: 70
Coordinates:
    South = 20
    North = 50
    East  = 20
    West  = 50
Using script: cloud_cloudsat_era_stats

Running sequence...
Running [level = 200 , diff = 70%]
Running [level = 250 , diff = 70%]
Running [level = 400 , diff = 70%]
Searching for min in
'/path/to/Cirrus/results/cloudsat/cloud_cloudsat_era_stats/stats_sat_2006_06_20
_2006_06_23_16_200mb_21_21_19.5_19.5.txt
Searching for min in
'/path/to/Cirrus/results/cloudsat/cloud_cloudsat_era_stats/stats_diff_2006_06_2
0_2006_06_23_16_200mb_21_21_19.5_19.5.txt
Searching for min in
'/path/to/Cirrus/results/cloudsat/cloud_cloudsat_era_stats/stats_sat_2006_06_20
_2006_06_23_16_250mb_21_21_19.5_19.5.txt
Searching for min in
'/path/to/Cirrus/results/cloudsat/cloud_cloudsat_era_stats/stats_diff_2006_06_2
0_2006_06_23_16_250mb_21_21_19.5_19.5.txt
Searching for min in
'/path/to/Cirrus/results/cloudsat/cloud_cloudsat_era_stats/stats_sat_2006_06_20
_2006_06_23_16_400mb_21_21_19.5_19.5.txt
Searching for min in
'/path/to/Cirrus/results/cloudsat/cloud_cloudsat_era_stats/stats_diff_2006_06_2
0_2006_06_23_16_400mb_21_21_19.5_19.5.txt
Sequence completed successfully!
```