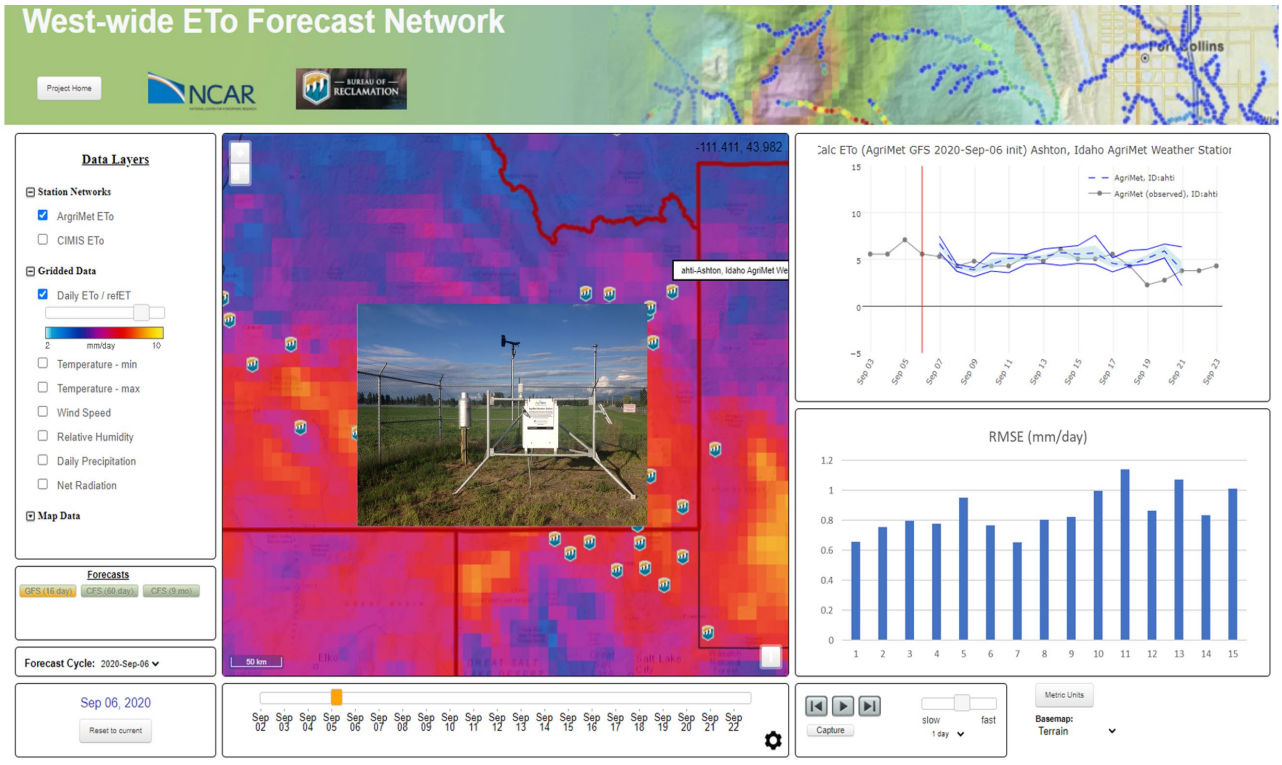
BUREAU OF RECLAMATION

The West-wide Evapotranspiration Forecast (WwET4Cast) Network

Science and Technology Program
Research and Development Office
Final Report No. ST-2021-1763-01



U.S. Department of the Interior

September 30, 2021

Mission Statements

The Department of the Interior (DOI) conserves and manages the Nation's natural resources and cultural heritage for the benefit and enjoyment of the American people, provides scientific and other information about natural resources and natural hazards to address societal challenges and create opportunities for the American people, and honors the Nation's trust responsibilities or special commitments to American Indians, Alaska Natives, and affiliated island communities to help them prosper.

The mission of the Bureau of Reclamation is to manage, develop, and protect water and related resources in an environmentally and economically sound manner in the interest of the American public.

Disclaimer

Information in this report may not be used for advertising or promotional purposes. The data and findings should not be construed as an endorsement of any product or firm by the Bureau of Reclamation, Department of Interior, or Federal Government. The products evaluated in the report were evaluated for purposes specific to the Bureau of Reclamation mission. Reclamation gives no warranties or guarantees, expressed or implied, for the products evaluated in this report, including merchantability or fitness for a particular purpose.

Additional Information

This document describes the software components associated with the S&T project 1763. This document as well as the WwET4Cast codes are present as items in the Reclamation Information Sharing Environment (RISE) user interface associated with Catalog ID 4615.

The West-wide Evapotranspiration Forecast (WwET4Cast) Network

Software Components Documentation

prepared for

Final Report No. ST-2021-1763-01

prepared by

California Great Basin Regional Office
Division of Planning
Decision Support Analysis Branch
Sacramento, California

Michael Tansey, Regional Climate Change Coordinator
Sacramento, California

David Yates, Scientist III
National Center for Atmospheric Research
Boulder, Colorado

Description of the ETo Forecasting System Software Components

The West-Wide ETo-Forecasting system is deployed within the Linux Operating System environment, and consists of a collection of software scripts, written in various interpreted language, including R, python, NCAR Command Language (NCL), bash shell, c-shell, and others. These various scripts are then cast within the System for Hydrologic Analysis, Research, and Prediction (SHARP) that is designed to facilitate the demonstration and evaluation of real-time forecasting workflows for short, medium and seasonal range predictions of both meteorological and hydrologic variables such as ETo and streamflow. The SHARP environment is controlled by the *ecFlow* workflow manager. The *ecFlow* is a client/server workflow package that allows for the execution of a large number of programs (with dependencies on each other and on time) in a controlled environment[1]. It provides reasonable tolerance for hardware and software failures, combined with restart capabilities, making it an ideal tool for deploying an operational forecasting system. It is important to note that an operational system such as SHARP is not immediately amenable to a typical software distribution, such as the typical installation of a single applications program, where there is usually a simple "setup.exe" type of program. Instead, this forecasting system includes a broad collection of individual codes and scripts, which when combined within the *ecFlow* environment, results in a robust system for managing and controlling forecasts.
The *ecFlow* workflow "Suite" is at the heart of the ETo forecasting system, as it controls the execution and timing of the individual scripts and programs that download necessary data, post-process those data into the forecasting environment, and then trigger various forecasting actions. Each *ecFlow* suite is controlled by the *ecFlow* server, which is deployed on the hydro-c1 cluster at NCAR. If there were interest in the USBR in setting up the forecasting system, it would be necessary to find a dedicated Linux server, where the *ecFlow* system could be installed, along with the unique SHARP components.

A subset of the *ecFlow* suite definition script, unique to the SHARP system, is shown below, with a focus on those components used for generating the ETo forecasts. An eCflow definition file is generally grouped into Families, where a Family generally addresses a particular set of sub-tasks, such as data acquisition, data processing, data post-processing, etc[2].
Each entry in the definition suite that begins with the **task** keyword identifies a specific action and with a particular timing for the execution of that task. Each *ecFlow* task within the Suite is controlled by a unique *.ecf* file that includes the scripts or programs that are executed for that particular task. For example, in the forecast suite below, the *task get_gefs* entry in the suite definition fie corresponds to a get_gefs.ecf file (described below the suite definition). The bolded entries are those specific to the ETo forecasting system, while the non-bolded tasks are part of the broader SHARP system.

[1] see https://confluence.ecmwf.int/display/ECFLOW/

[2] See https://confluence.ecmwfint/display/ECFLOW/Suite+Definitions for details

```
#############################################################################################
#4.0.7. eCflow version
Suite forecast
family main
    family data_acquisition
      task get_ghcnd
        time 12:00        # check to see if files appear earlier
      task get_nwcc       # ditto check to see if files appear earlier
        trigger /forecast/main/data_acquisition/get_ghcnd eq complete
        edit SNOTEL_OUTFILE '/d3/hydrofcst/overtheloop/data/stn_obs/SNOTEL/nwcc_download.csv'
      task get_gefs
        time 17:00
      task get_cfsr
        edit CFSR_CFG '/home/hydrofcst/SHARP/sharp_realtime/config/op/get_cfsr.cfg'
        time 20:00
      task get_gfs
        time 17:00
      task get_cfs
        time 17:00
      task get_obs_flows
        edit GET_FLOW_CFG '/home/hydrofcst/SHARP/sharp_realtime/config/op/get_flow.cfg'
        time 15:00
      task get_RFC_flows
         #trigger /forecast/main/initial_conditions/spinup_nws/run_nws_spinup eq complete
         edit RFC_FLOW_CFG '/home/hydrofcst/SHARP/sharp_realtime/config/op/get_RFC_flows.cfg'
         time 18:00
      task get_NWM_flows
        time 20:00  # try 2 pm in the afternoon
        # time 15:00 22:00 01:00 # every hour from 15:00 to 22:00
        edit NWM_FLOW_CFG '/home/hydrofcst/SHARP/sharp_realtime/config/op/get_NWM_flows.cfg'
      task get_et_met
        time 10:00
        edit ETMET_CFG '/home/hydrofcst/SHARP/sharp_realtime/config/op/get_et_met.cfg'
      task get_et_cimis
        time 10:00
        edit CIMIS_CFG '/home/hydrofcst/SHARP/sharp_realtime/config/op/get_cimis_met.cfg'
      task get_shootout_flows
        time 13:20 16:20 00:30
      task get_ecmwf
       time 10:00
    endfamily

    family data_reformat
      task reformat_ghcnd
        trigger /forecast/main/data_acquisition/get_ghcnd eq complete
      task reformat_nwcc
        trigger /forecast/main/data_acquisition/get_nwcc eq complete and
/forecast/main/data_reformat/reformat_ghcnd eq complete
      task reformat_gfs
        trigger /forecast/main/data_acquisition/get_gfs eq complete
      task reformat_cfs
        trigger /forecast/main/data_acquisition/get_cfs eq complete
      task reformat_cfs_monthly
        trigger /forecast/main/data_reformat/reformat_cfs eq complete
      task reformat_cfs_60day
        trigger /forecast/main/data_reformat/reformat_cfs eq complete
      task reformat_cfs_9month
        trigger /forecast/main/data_reformat/reformat_cfs_monthly eq complete
    endfamily

    family mr_forecast
     family et4cast
        task gefs2asc_etvars
          trigger /forecast/main/data_acquisition/get_gefs eq complete
          edit CFG '/home/hydrofcst/SHARP/sharp_realtime/config/op/gefs2asc_etvars.points.cfg'
        task gefsAndAgrimet_etvars
          trigger /forecast/main/mr_forecast/et4cast/gefs2asc_etvars eq complete and
/forecast/main/data_acquisition/get_et_met eq complete
          edit CFG
'/home/hydrofcst/SHARP/sharp_realtime/config/op/gefsAndAgrimet_etvars.points.cfg'
        task gefs2asc_cimis
```

```
          trigger /forecast/main/data_acquisition/get_gefs eq complete
          edit CFG '/home/hydrofcst/SHARP/sharp_realtime/config/op/gefs2asc_cimis.points.cfg'
        task gefsAndCIMIS_etvars
          trigger /forecast/main/mr_forecast/et4cast/gefs2asc_cimis eq complete and
/forecast/main/data_acquisition/get_et_cimis eq complete
          edit CFG
'/home/hydrofcst/SHARP/sharp_realtime/config/op/gefsAndCIMIS_etvars.points.cfg'
        task gfs2asc_agrimet
          trigger /forecast/main/data_acquisition/get_gfs eq complete
          edit CFG '/home/hydrofcst/SHARP/sharp_realtime/config/op/gfs2asc_agrimet.points.cfg'
        task gfsAndAgrimet_etvars
          trigger /forecast/main/mr_forecast/et4cast/gfs2asc_agrimet eq complete and
/forecast/main/data_acquisition/get_et_met eq complete
          edit CFG
'/home/hydrofcst/SHARP/sharp_realtime/config/op/gfsAndAgrimet_etvars.points.cfg'
        task gfs2asc_cimis
          trigger /forecast/main/data_acquisition/get_gfs eq complete
          edit CFG '/home/hydrofcst/SHARP/sharp_realtime/config/op/gfs2asc_cimis.points.cfg'
        task gfsAndCIMIS_etvars
          trigger /forecast/main/mr_forecast/et4cast/gfs2asc_cimis eq complete and
/forecast/main/data_acquisition/get_et_cimis eq complete
          edit CFG '/home/hydrofcst/SHARP/sharp_realtime/config/op/gfsAndCIMIS_etvars.points.cfg'
        task GriddedGFS_ET
          trigger /forecast/main/data_acquisition/get_gfs eq complete
        task GriddedGEFS_ET
          trigger /forecast/main/data_acquisition/get_gefs eq complete
      endfamily
    endfamily
  endfamily
endsuite
```

Within the *ecFlow* work directory on the Hydro-C1 cluster, the directory that contains the *ecFlow* SHARP "forecast" suite is *$ECFLOW*[3]*/sharp_realtime/* and within this directory are sub-directories that contain each *ecFlow* family that encapsulate those unique functionalities. For example, the *data_acquisition* family contains the controlling scripts for downloading various datasets, including the USBR Agrimet network data and CIMIS data, using the *get_et_met* and the *get_et_cimis* ecFlow tasks. Also within the *data_acquisition* family is a task for downloading global forecast system data, which is identified on line 9 in the above Suite definition file. The actual *ecFlow* script is *get_gefs.ecf* which itself, identifies a python script and configuration file for downloading Global Ensemble Forecast System (GEFS) data. These scripts then converts the data into a NetCDF format and store those outputs on the Hydro-C1 data server. The *get_gefs.ecf* script includes syntax specific to its control on the Hydro-C1 cluster (the #SBATCH controls), as the SHARP system is managed within a SLURM[4] workload manager, which itself has a unique set of commands and tools for controlling the submission of jobs across a distributed, multi-user platform. The *get_gefs.efs* script is shown below, where the actual downloading and processing code is within a python script, "get_gefs.py".

```
%include <qsub.h>
# --- job-specific settings ---
#SBATCH --partition=main
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --time=2:00:00
#SBATCH --output=%LOG_DIR%/get_gefs.log
%include <head.h>
%include <noncritical_errs.h>
%manual
This script downloads Global Ensemble Forecast System (GEFS) data from
ftp://ftp.cdc.noaa.gov/Projects/Reforecast2/ using wget. Converts output data to netcdf format.
```

[3] $ECFLOW – this is an environment variable, which is used to define the overall location of the ecFlow installation, which will be unique to the installation.

[4] Slurm is an open source, fault-tolerant, and highly scalable cluster management and job scheduling system for large and small Linux clusters.

```
%end
echo submit `date`
cd %DATA_STREAMS%/acquisition/
python get_gefs.py %CFG_FILE%
echo submit `date`
```

The next family in the definition suite is the *data_reformat* family, which includes a set of scripts and programs for post-processing various climate model data output. Of particular interest for the ETo forecasting project is the processing of the Climate Forecast System (CFS) forecasts and the development of meteorological forecasts at 60-day and 9-month time horizons that are needed for generating a long-range ETo forecast. An example *ecFlow* tasks in this family is the *reformat_cfs_60day*. The *eCflow* task script, shown below, invokes a simple csh script, which resides within the reformat directory of the *eCflow* directory suite- *cfs_make-60day-ensemble.csh*.

```
%include <qsub.h>
# --- job-specific settings ---
#SBATCH --partition=main
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=4
#SBATCH --time=5:00:00
#SBATCH --output=%LOG_DIR%/reformat_cfs_60day.log

%include <head.h>
%include <noncritical_errs.h>
%manual
This script makes dailies products out of CFS forecast hourly outputs
%end
cd %DATA_STREAMS%/reformat/
echo submit `date`
./cfs_make-60day-ensemble.csh
echo end `date`
%include <tail.h>
```

The *cfs_make-60day-ensemble.csh* script uses NCO operators and NCL scripts to reformat and post-process the CFS data and generate an ETo forecast at the 60-day time horizon. It does this using a lagged ensemble, where multiple forecasts made on the previous day are used to generate a 60-day ETo ensemble forecast, each day. The details of this script are shown below. Note that we are showing this *ecFlow* tasks script to highlight the flexibility of the forecasting system in terms of adding and modifying certain processes, and taking advantage of particular tools and approaches. In this case, NCO and NCL are used together, to generate the CFS 60-day forecast. A drawback of this type of system is that all of the pre-requisite tools must be deployed on the server. For example, both NCO and NCL must be installed for this script to function. The *cfs_make-60day-ensemble.csh:*

```
#!/bin/tcsh
if  ($#argv == 3) then
        echo "got date"
        set year        = $argv[1]
        set month       = $argv[2]
        set d           = $argv[3]
        set day         = $argv[1]$argv[2]$argv[3]
        set YM          = $argv[1]$argv[2]
else
        echo "determine date automatically : today"
        set year        = `date +%Y`
        set month       = `date +%m`
        set d           = `date +%d`
        set day         = `date +%Y%m%d`
        set YM          = {$year}{$month}
endif
setenv  project        'et'
setenv  cfsyear        $year
setenv  cfsmonth       $month
setenv  cfsday         $d
set     indir   =   /d3/hydrofcst/overtheloop/data/cfs/$year/$YM/$day
echo    $indir
```

```
##
## compute daily and monthly annual cycles with distribution information
#module load ncl
ncl cfs_make-60day-ensemble.ncl;
wait
##
## cleanup and combine variables into single files for daily and monthly-stats
##
set varlist = ( "tas" "tasmin" "tasmax" "pr" "ps" "rhum" "netrad" "windspeed" "u10" "v10")
rm -rf {$indir}/cfsv2_et*_60-day-ensemble.nc
set outfile = `echo {$indir}/refevt_{$project}_*_60-day-ensemble.nc | sed "s/refevt/cfsv2/g"`
cp {$indir}/refevt_{$project}_*_60-day-ensemble.nc $outfile
foreach var($varlist)
        ncks -h -A {$indir}/{$var}_{$project}_*_60-day-ensemble.nc  $outfile
end
rm -rf $indir/refevt_{$project}_*_60-day-ensemble.nc
foreach var($varlist)
        rm -rf $indir/{$var}_{$project}_*_60-day-ensemble.nc
end
```

The *et4cast* Family is the main *eCflow* family for generating station-based, ETo forecast. These tasks include:

- *gefs2asc_etvars*- This scripts extracts and formats sub-daily GEFS forecast data into daily data for the individual Agrimet stations. In the ecFlow workflow process, this script is triggered upon completion of the downloading of the GEFS forecast data, and is dependent upon the configuration file given below.

  trigger /forecast/main/data_acquisition/get_gefs eq complete
  edit CFG '/home/hydrofcst/SHARP/sharp_realtime/config/op/gefs2asc_etvars.points.cfg'

- *gefsAndAgrimet_etvars*- This script bias corrects the meteorological fields of the GEFS forecasts for use in the ETo forecast estimate for the Agrimet stations.

- *gefs2asc_cimis* - This scripts extracts and formats sub-daily GEFS forecast data into daily data for the individual CIMIS stations

- *gefsAndCIMIS_etvars* - This script bias corrects the meteorological fields of the GEFS forecasts for use in the ETo forecast estimate for the CIMis stations.

- *gfs2asc_agrimet* - This scripts extracts and formats sub-daily the GFS forecast data into daily data for the individual Agrimet stations

- *gfsAndAgrimet_etvars* - This script bias corrects the meteorological fields of the GFS forecasts for use in the ETo forecast estimate for the Agrimet stations for generating the 16 days forecasts.

- gfsAndCIMIS_etvars - This script bias corrects the meteorological fields of the GFS forecasts for use in the ETo forecast estimate for the Agrimet stations.

- GriddedGFS_ET – This script generates a gridded ETo forecast from the raw GFS model forecast ouputput, allowing for a ETo forecast anywhere in the domain, out 16 days.

- GriddedGEFS_ET – This script generate an ETo forecast from the

An example of the ecFlow control for the gefs2as_etvars.ecf is shown below, where the actual script for generating the forecast is an R script,

```
# --- job-specific settings ---
#SBATCH --partition=main
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --time=1:00:00
#SBATCH --output=%LOG_DIR%/gefs2asc_etvars.log
%include <head.h>
%include <critical_errs.h>
%manual
This script extracts and processes a list of variables from the GEFS forecast data
for selected points.  The data are designed to be used for an ET estimation model.
%end
cd %METHODS%/et4cast/
echo submit `date`
StartDate=$(date +%%Y%%m%%d --date="yesterday")
./gefs2asc_etvars.points.Rscr %CFG% $StartDate
echo end `date`
```

The .cfg control script (defined in the *ecFlow* command above using the **%CFG%** environment variable) is given below, and demonstrates the use of the R scripting language as one of the tools used in the overall forecasting process. The script controls the stations used, the location of the input and output data, the naming convention of the stations, the forecast ensembles used, the available meteorological output variables, the type of output variables, the aggregation type, and the number of ensembles members to save. The other scripts within the *et4cast* system use a similar approach and can generally be found in the same location as this script.

```
# ==== configuration file for gefs2asc_etvars.points.Rscr ===
Region        <- "XCONUS"      # used in file paths
Project       <- "et_points"
PointInfo     <- "/home/hydrofcst/SHARP/sharp_realtime/info/metadata.et_sites.v0.csv"
InputFilePath <- "/d3/hydrofcst/overtheloop/data/gefs2/"
OutputPath    <- "/d3/hydrofcst/et4cast/data/model/gefs_etvars"
NSigOut       <- 2 # number of sig digits on output data
PointID       <- c("ALL")  # c("ALL")    # can also specify individual labels
FcstEnsLbl    <- c("c00","p01","p02","p03","p04","p05","p06","p07","p08","p09","p10","mean")
GEFSVars      <- c("tmp_2m","apcp_sfc","dlwrf","dswrf","pres_msl","ugrd_80m","vgrd_80m","spfh")
GEFSVarAggr   <- c("mean","sum","mean","mean","mean","mean","mean","mean")
OutputType    <- "ETVarInputs"  #"ETVarInputs" or "ETVars"
AggToDaily    <- TRUE
NoEns         <- FALSE  # TRUE:single file run (InputFileRoot will be used as a filename)
```