FV3GFS Workflow How-To

How to clone, build, set up and run the FV3GFS with Rocoto

Kate Friedman - NCEP/EMC EIB

FV3GFS Workflow How-To - Table of Contents

The general sections of this document (click link to jump to section):

Document change information - v3.7

Author: Kate (Howard) Friedman (kate.friedman@noaa.gov)

Version 3.7 (June 14th, 2018)

- Added slides documenting common errors seen and resolutions.
- Added “What’s coming?!” slide for big changes in the works.
- Added table of contents slide at start of document with links that jump to each section.

Version 3.6 (May 21st, 2018)

- Add slide explaining how FV3GFS connects to its components/sub-modules.

Version 3.5 changes (April 4th, 2018)

- Added slide about job sequence.
- Added slide detailing what each job in sequence does.

See final slides for older updates to document.

Important pages and documentation

- FV3GFS home on VLab: https://vlab.ncep.noaa.gov/group/fv3gfs

- FV3GFS workflow Redmine page:
  - https://vlab.ncep.noaa.gov/redmine/projects/fv3gfs

- FV3GFS sub-module Redmine pages:
  - NEMSfv3gfs (POC: Sam Trahan): https://vlab.ncep.noaa.gov/redmine/projects/nemsfv3gfs
  - comfv3 (POC: Jun Wang): https://vlab.ncep.noaa.gov/redmine/projects/comfv3
  - GSI (POC: Mark Potts): https://vlab.ncep.noaa.gov/redmine/projects/comgsi
  - UPP (POCs: Wen Meng/Huiya Chuang): https://vlab.ncep.noaa.gov/redmine/projects/emc-post

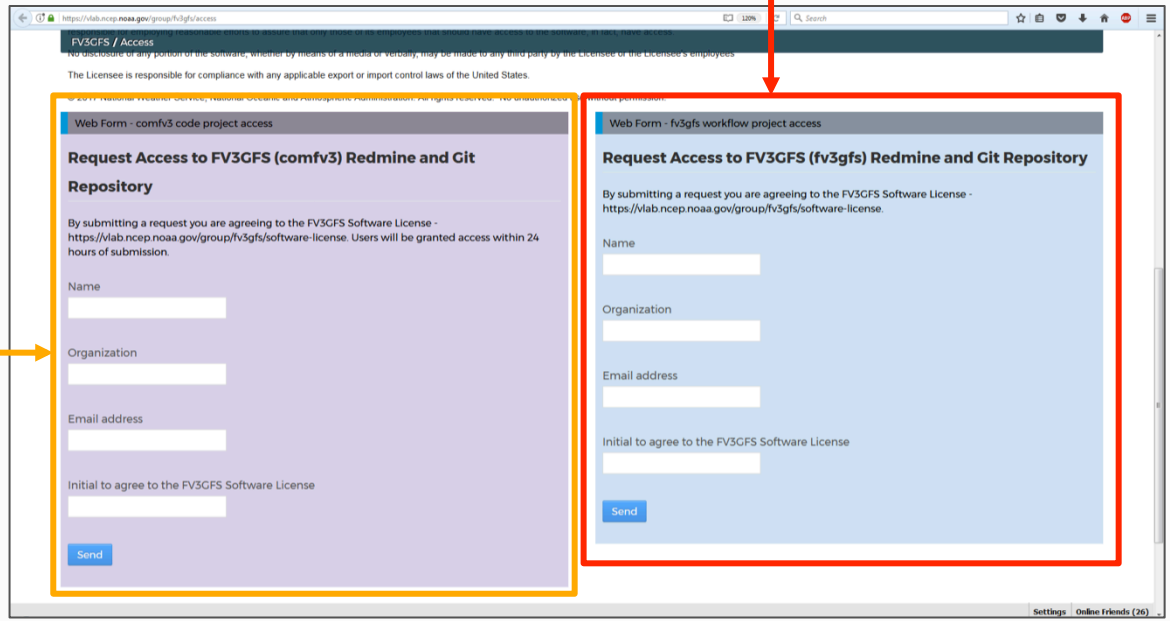
- CROW VLab community (see what the CROW team is working on):
  - https://vlab.ncep.noaa.gov/group/crow

Gain access to ‘fv3gfs’ project on VLDS - needed to clone fv3gfs git repository

Visit FV3GFS VLab community ‘Access’ page and submit forms for comfv3 and fv3gfs project access:
https://vlab.ncep.noaa.gov/group/fv3gfs/access

If you haven’t requested access to the fv3gfs sub-module projects yet do so now as well. They are needed to checkout component codes within the workflow:

- **NEMSfv3gfs**: Sam Trahan (click here for NEMSfv3gfs form)
- **comfv3**: Jun Wang, see form
- **GSI:** Mark Potts (email for access)
- **Post/UPP**: Wen Meng & Huiya Chuang (click here for post/UPP form)



Working Status of FV3GFS Git Master

As of June 14th, 2018:

Mode	Platform	Status	Notes
Free-forecast	WCOSS-Cray	✔	All is well.
	Theia	✔	
Cycling	WCOSS-Cray	✔	
	Theia	✔	

Report run issues to Fanglin Yang (fanglin.yang@noaa.gov) and Kate Friedman (kate.friedman@noaa.gov)

Major changes in the works - what's coming?!

- Porting FV3GFS to WCOSS-Dell.
- Porting FV3GFS to Jet and Gaea.
- Developing new front-end using the Community Research and Operational Workflows (CROW) system.

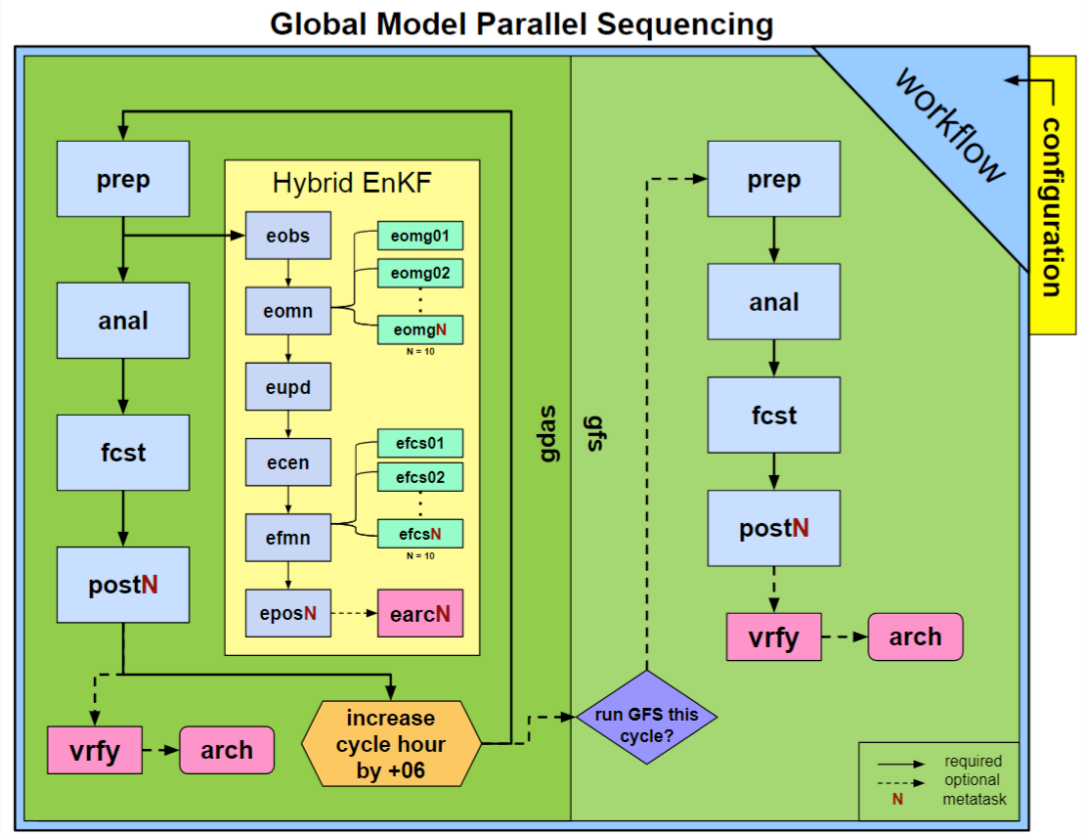
The FV3GFS Job Sequence

FV3GFS experiments employ the global model parallel sequencing (shown to the right). The system utilizes a collection of scripts that perform the tasks for each step.

As with the operational system, the **gdas** provides the guess fields for the **gfs**. The **gdas** runs for each cycle (00, 06, 12, and 18 UTC), however, to save time and space in experiments the **gfs** (right side of the diagram) is initially setup to run for only the 00 UTC cycle. (See the "run **GFS** this cycle?" portion of the diagram) The option to run the **GFS** for all four cycles is available (see gfs_cyc variable in configuration file).

An experimental run is different from operations in the following ways:

- Dump step is not run as it has already been completed during the real-time production runs and dump data is available in the global dump archive on supported machines.
- Addition steps in experimental mode:
  - verification (vrfy)
  - archive (arch)



Flow diagram of a typical experiment with Hybrid EnKF turned ON

The FV3GFS Job Sequence - what each job does

anal	Runs the analysis. 1) update surface guess file via global_cycle to create surface analysis; 2) runs the atmospheric analysis (global_gsi); 3) updates the angle dependent bias
arch/earc	(arch) Archives select files and cleans up older data. (earc) Archival script for Hybrid EnKF: 1) Write select EnKF output to HPSS; 2) Copy select files to online archive; 3) Clean up EnKF temporary run directories; 4) Remove "old" EnKF files from rotating directory.
ecen	Multiple functions: 1) Compute ensemble mean analysis from 80 analyses generated by eupd; 2) Perturb 80 ensemble analyses; 3) Compute ensemble mean for perturbed analyses; 4) Chgres high resolution analysis to ensemble resolution; 5) Recenter perturbed ensemble analysis about high resolution analysis.
efcs	Run 9 hour forecast for each ensemble member. There are 80 ensemble members. Each efcs job sequentially processes 8 ensemble members, so there are 10 efcs jobs in total.
eobs	Run GSI to select observations for all ensemble members to process. Data selection done using ensemble mean.
eomg	Compute innovations for ensemble members. Innovations computed by running GSI in observer mode. It is an 80 member ensemble so each eomg job sequentially processes 8 ensemble members.
epos	Compute ensemble mean surface and atmospheric mean ensemble files.
eupd	Perform EnKF update (i.e., generate ensemble member analyses).
fcst	Runs the forecast.
prep	Runs the data preprocessing prior to the analysis (storm relocation if needed and generation of prepbufr file).
post	Runs the post processor.
vrfy	Runs the verification step.

The FV3GFS Component Relationship and Repository Connections

The FV3GFS is a combination of several modeling components working together to prepare, analyze, produce, and post-process forecast data.

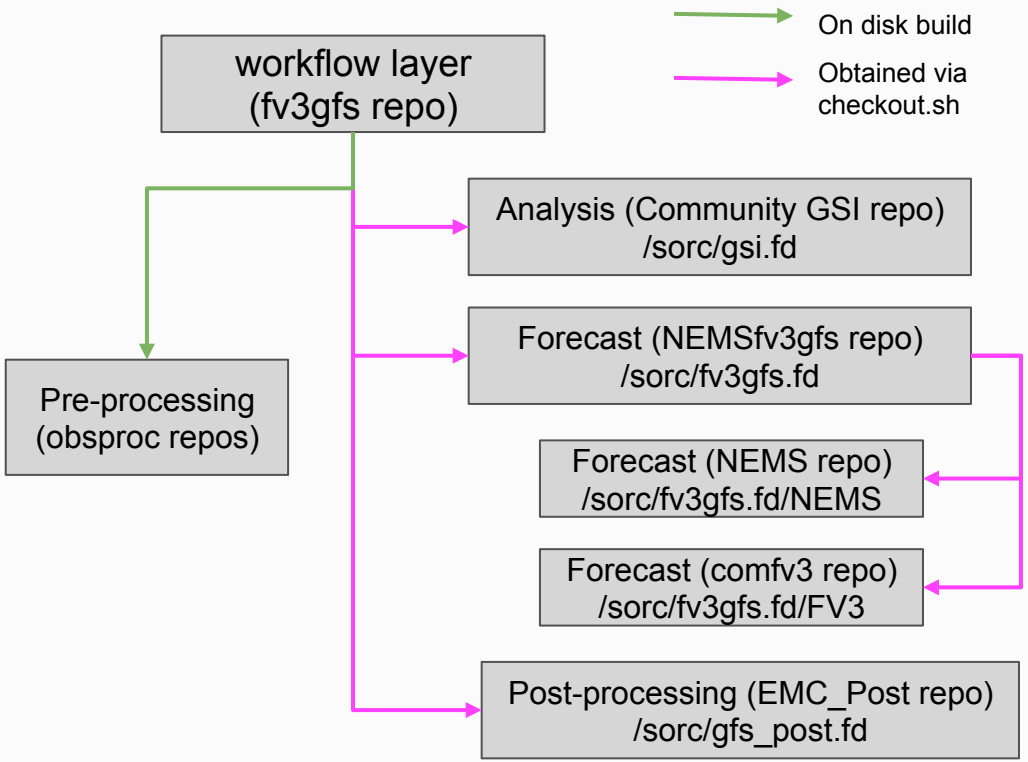
The major components of the system are:

- Workflow
- Pre-processing
- Analysis
- Forecast
- Post-processing

The FV3GFS ('fv3gfs') repository contains the workflow layer and, after running the checkout script, the code and scripts for the analysis, forecast, and post-processing components. Any non-workflow component is known as a sub-module. All of the sub-modules of the system reside in their respective repositories on the NWS Virtual Lab server, which are managed using the Redmine and Gerrit code management tools.

The diagram to the right provides a depiction of how the various components of the system connect to work together via their respective repositories on VLab. The FV3GFS sub-modules are obtained by running the checkout script found under the /sorc folder. More on this in a later slide.

Additional utilities, codes, and scripts used by the system also reside in the workflow repo.



How to set up and run the FV3GFS w/ Rocoto

1. Clone workflow
2. Build system
3. Set up run
4. Start run
5. Monitor run
6. Additional instructions and utilities

Clone & Checkout Branch

Prior to cloning make sure your gerrit access is set up. *Click here to view VLab presentation with instructions.*

> git clone --recursive gerrit:fv3gfs
Alternative clone command: > git clone ssh://$USER@vlab.ncep.noaa.gov:29418/fv3gfs ← $USER = first.last
> cd fv3gfs
> git branch ← not required, command will show what you have checked out
* master ← by default you now have the master only

You now have a cloned copy of the fv3gfs git repository, which currently includes a copy of the master.
To checkout a branch or tag in your clone:

> git checkout $BRANCH_NAME

The “checkout” command will checkout $BRANCH_NAME and switch your clone to that branch.
Example:

> git checkout my_branch ← checkout the ‘my_branch’ branch into clone
> git branch
* my_branch ← now your clone is the ‘my_branch’ branch
  master

Next slide for build instructions...

Build codes under /sorc

> cd sorc (***following steps are all required***)

> sh checkout.sh ← checks out fv3gfs.fd, gsi.fd, and gfs_post.fd codes

> sh build_all.sh ← for partial build modify sorc/fv3gfs_build.cfg, then run build_all.sh

Individual build_*.sh scripts can be run separately as needed (e.g. sh build_fv3.sh).

> sh link_fv3gfs.sh emc[nco] cray[theia] ← defau

NCO (don't use)

```
LUNA-llogin2 > ls ../exec/
calc_increment_ens.x      gfs_ncep_post          oznmon_time.x
chgres_recenter.exe       global_chgres          radmon_angle
cnvgrib21_gfs             global_cycle           radmon_bcoef
emcsfc_ice_blend          global_enkf            radmon_bcor
emcsfc_snow2mdl           global_gsi             radmon_time
fbwndgfs                  gridbull               recentersigp.x
filter_topo               make_hgrid             regrid_nemsio
fregrid                   make_hgrid_parallel    relocate_mv_nvortex
fregrid_parallel          make_solo_mosaic       shave.x
fv3_gfs_nh.prod.32bit.x   mkgfsnemsioctl         supvit
fv3nc2nemsio.x            ml01rg2.x              syndat_getjtbul
gaussian_sfcanl.exe       navybull               syndat_maksynrc
gdas_trpsfcmv             nemsio_cvt             syndat_qctropcy
getsfcensmeanp.x          nemsio_get             tave.x
getsigensmeanp_smooth.x   nemsio_read            tocsbufr
getsigensstatp.x          nst_tf_chg.x           vint.x
gettrk                    overpdtg2              wintemv
gfs_bufr                  oznmon_horiz.x
```

```
LUNA-llogin2 > sh build_all.sh
Creating logs folder
Creating ../exec folder
 .... Building fv3 ....
 .... Building gsi ....
 .... Building ncep_post ....
 .... Building NEMS util ....
 .... Building chgres ....
 .... Building gaussian_sfcanl and nst_tf_chg ....
 .... Building orog ....
 .... Building cycle ....
 .... Building enkf_chgres_recenter ....
 .... Building tropcy_NEMS ....
 .... Building gdas ....
 .... Building gfs_fbwndgfs ....
 .... Building gfs_overpdtg2 ....
 .... Building gfs_wintemv ....
 .... Building gfs_cnvgrib21_gfs ....
 .... Building gfs_bufrsnd ....
 .... Building emcsfc ....
 .... Building fre-nctools ....
 .... Building fv3nc2nemsio ....
 .... Building regrid_nemsio ....

 .... Build system finished ....
```

FV3GFS Initial Conditions

Cycled experiments:

- Initial conditions currently set up for use on WCOSS_C and Theia:
  - C768 deterministic & C384 ensemble (Cray only, ops-like, resource hog, use sparingly!)
    - idates: 2018010500, 2018012306
  - C384 deterministic & C192 ensemble *(default supported settings)*
    - idates: 2015113000, 2016100100, 2017073118, 2018010500
  - C192 deterministic & C96 ensemble: C96 available for 2017073118 only, C192 det/ens for 2018010500
- Paths (--icsdir):
  - Theia: /scratch4/NCEPDEV/global/noscrub/glopara/ICS/FV3GFS
  - WCOSS_C: /gpfs/hps3/emc/global/noscrub/emc.glopara/ICS

Forecast-only experiments:

- Place pre-prepared ICs in folders with the following format: $ICSDIR/$CDATE/gfs/C$RES/INPUT
- Set ICSDIR in config.base prior to running setup_workflow_fcstonly.py. Learn more about setup scripts in following slides.

Operational NEMS/GSM output for making FV3 ICs on HPSS: /NCEPPROD/hpssprod/runhistory/rhYYYY/YYYYMM/YYYYMMDD

- cycled: gpfs_hps_nco_ops_com_gfs_prod_gdas.YYYYMMDDCC.tar
- cycled enkf: gpfs_hps_nco_ops_com_gfs_prod_enkf.YYYYMMDD_CC.anl.tar
- free-forecast gfs: gpfs_hps_nco_ops_com_gfs_prod_gfs.YYYYMMDDCC.anl.tar

Next slide for instructions on generating your own initial conditions...

FV3GFS Initial Conditions - Making your own

If you **do not wish to use the canned ICs** (i.e. use different source or date), or **need to prepare ICs for free-forecasts**, then you can run the following script to convert GFS ICs to FV3 ICs. (Note: this script is only one-way, GFS->FV3)

ush/rocoto/fv3ics.py

Usage:

python fv3ics.py --date $CDATE --icsdir $ICSDIR --CASE_det $RES_DET --CASE_ens $RES_ENS

- $CDATE is the date of your ICs (YYYYMMDDCC)
- $ICSDIR is the location of your ICs to convert. See right for the expected format of ICSDIR.
- $RES_DET is the resolution of your converted deterministic FV3 ICs (do not include “C”)
- $RES_ENS is the resolution of your converted ensemble FV3 ICs (do not include “C”)

The fv3ics.py script expects the following ICSDIR format ⟶

Input files to convert - Resulting FV3GFS ICs after convert

$ICSDIR/

- $CDATE/ (YYYYMMDDHH)

- gdas.tHHz.abias

- gdas.tHHz.abias_pc

- gdas.tHHz.abias_air

- gdas.tHHz.radstat

- T1534/ (JCAP_det, typically from operations)

- gdas.tHHz.atmanl.nemsio

- gdas.tHHz.sfcanl.nemsio

- gdas.tHHz.nstanl.nemsio (optional)

- T574/ (JCAP_ens, typically from operations)

- gdas.tHHz.ratmanl.memXXX.nemsio

- gdas.tHHz.sfcanl.memXXX.nemsio

- gdas.tHHz.nstanl.memXXX.nemsio (optional)

- C###/ (CASE_det, CASE_ens, i.e. C96 or C384)

- control (CASE_det output)

- fv3ics.log (chgres log for control)

Set up experiment - part 1a - cycled experiment

Run setup_expt.py script to create EXPDIR and COMROT:

> cd ush/rocoto
> ./setup_expt.py --pslot $PSLOT --configdir $CONFIGDIR --idate $IDATE --edate $EDATE --icsdir $ICSDIR --comrot $COMROT --expdir $EXPDIR [--resdet $RESDET --resens $RESENS --nens $NENS --gfs_cyc $GFS_CYC]

- $PSLOT is the name of your experiment
- $CONFIGDIR is the path to the /config folder under the copy of the system you're using (i.e. ../parm/config/)
- $IDATE is the initial start date of your run (first cycle CDATE, YYYYMMDDCC)
- $EDATE is the ending date of your run (YYYYMMDDCC) and is the last cycle that will complete
- $ICSDIR is the path to the ICs for your run. See prior slide for paths.
- $COMROT is the path to your experiment output directory. Do not use noscrub space on Cray for COMROT, use ptmp.
  DO NOT include PSLOT folder at end of path, it'll be built for you.
- $EXPDIR is the path to your experiment directory where your configs will be placed and where you will find your workflow monitoring files (i.e. rocoto database and xml file). *DO NOT include PSLOT folder at end of path*, it will be built for you.
- $RESDET is the resolution of the deterministic forecast (i.e. '--resdet 768', optional, default is C384)
- $RESENS is the resolution of the ensemble (EnKF) forecast (i.e. '--resens 384', optional, default is C192)
- $NENS is the number of ensemble members (optional, default is 20)
- $GFS_CYC is the cycle frequency of the long GFS forecast (0 = none, 1 = 00z only [default], 2 = 00z & 12z, 4 = all cycles)

See later slide for forecast-only experiment setup information and examples.

Set up experiment - part 1a - cycled experiment

Example setup_expt.py on WCOSS_C:

SURGE-slogin1 > ./setup_expt.py --pslot fv3demo --configdir /gpfs/hps3/emc/global/noscrub/Joe.Schmo/git/fv3gfs/parm/config --idate 2017073118
--edate 2017080106 --icsdir /gpfs/hps3/emc/global/noscrub/emc.glopara/ICS --comrot /gpfs/hps2/ptmp/Joe.Schmo --expdir /gpfs/hps3/emc/global/noscrub/Joe.Schmo/para_gfs

SDATE = 2017-07-31 18:00:00
EDATE = 2017-08-01 06:00:00

EDITED: /gpfs/hps3/emc/global/noscrub/Joe.Schmo/para_gfs/fv3demo/config.base as per user input.
DEFAULT: /gpfs/hps3/emc/global/noscrub/Joe.Schmo/para_gfs/fv3demo/config.base.default is for reference only.
Please verify and delete the default file before proceeding.

SURGE-slogin1 >

---

*The message about the config.base.default is telling you that you are free to delete it if you wish but it's not necessary to remove. Your resulting config.base was generated from config.base.default and the default one is there for your information.

Set up experiment - part 1a - cycled experiment

What happens if I run setup_expt.py again for an experiment that already exists:

SURGE-slogin1 > ./setup_expt.py --pslot fv3demo --configdir /gpfs/hps3/emc/global/noscrub/Joe.Schmo/git/fv3gfs/parm/config --idate 2017073118
--edate 2017080106 --icsdir /gpfs/hps3/emc/global/noscrub/emc.glopara/ICS --comrot /gpfs/hps2/ptmp/Joe.Schmo --expdir /gpfs/hps3/emc/global/noscrub/Joe.Schmo/para_gfs

COMROT already exists in /gpfs/hps2/ptmp/Joe.Schmo/fv3demo

Do you wish to over-write COMROT [y/N]: y

EXPDIR already exists in /gpfs/hps3/emc/global/noscrub/Joe.Schmo/para_gfs/fv3demo

Do you wish to over-write EXPDIR [y/N]: y

SDATE = 2017-07-31 18:00:00
EDATE = 2017-08-01 06:00:00

EDITED: /gpfs/hps3/emc/global/noscrub/Joe.Schmo/para_gfs/fv3demo/config.base as per user input.
DEFAULT: /gpfs/hps3/emc/global/noscrub/Joe.Schmo/para_gfs/fv3demo/config.base.default is for reference only.
Please verify and delete the default file before proceeding.

Your COMROT and EXPDIR will be deleted and remade. Be careful with this!

Set up experiment - part 1b - forecast-only experiment

Run setup_expt_fcstonly.py script to create EXPDIR and COMROT:

> cd ush/rocoto

> ./setup_expt_fcstonly.py --pslot $PSLOT --configdir $CONFIGDIR --idate $IDATE --edate $EDATE --res $RES --gfs_cyc $GFS_CYC --comrot $COMROT --expdir $EXPDIR

- $PSLOT is the name of your experiment
- $CONFIGDIR is the path to the /config folder under the copy of the system you're using (i.e. ../parm/config/)
- $IDATE is the initial start date of your run (first cycle CDATE, YYYYMMDDCC)
- $EDATE is the ending date of your run (YYYYMMDDCC) and is the last cycle that will complete
- $RES is the resolution of the forecast (i.e. 768 for C768)
- $GFS_CYC is the forecast frequency (0 = none, 1 = 00z only [default], 2 = 00z & 12z, 4 = all cycles)
- $COMROT is the path to your experiment output directory. *DO NOT include PSLOT folder at end of path*, it'll be built for you.
- $EXPDIR is the path to your experiment directory where your configs will be placed and where you will find your workflow monitoring files (i.e. rocoto database and xml file). *DO NOT include PSLOT folder at end of path*, it will be built for you.

Set up experiment - part 2 - check and adjust base settings before proceeding

- Review the contents of your EXPDIR, you should have configs within it now.
- Open config.base in EXPDIR and adjust as needed (i.e. change EDATE to run a shorter/longer experiment). Definitely double check ACCOUNT, HPSS_PROJECT, HOMEDIR, STMP, PTMP, NOSCRUB, MYBASE_SVN variables.

  > vi $EXPDIR/$PSLOT/config.base

  If running forecast-only experiment change ICSDIR to your the location of your prepared ICs. See earlier slides for more details on preparing ICSDIR.
- Review the contents of your COMROT, you should have initial file folders setup with symlinks to initial conditions. More on this in next slide...

Set up experiment - part 2 - state of COMROT after running setup_expt.py

After running setup_expt.py your COMROT will be setup with initial conditions. The COMROT folder is divided into CDUMP.date folders and then further divided by cycle within. The workflow is currently setup to start a half cycle with the first jobs being gdas fcst and gdas efmn (EnKF forecast job).

```
SURGE-slogin1 > pwd
/gpfs/hps2/ptmp/Joe.Schmo/fv3demo

SURGE-slogin2 > ls -l
total 0
drwxr-xr-x 3 Joe.Schmo g01 512 Dec  5 17:37 enkf.gdas.20170731
drwxr-xr-x 3 Joe.Schmo g01 512 Dec  5 17:37 gdas.20170731

SURGE-slogin2 > ls -l gdas.20170731/18/
total 0
lrwxrwxrwx 1 Joe.Schmo g01 75 Dec  5 17:37 INPUT -> /gpfs/hps3/emc/global/noscrub/emc.glopara/ICS/2017073118/C384/control/INPUT
lrwxrwxrwx 1 Joe.Schmo g01 72 Dec  5 17:37 gdas.t18z.abias -> /gpfs/hps3/emc/global/noscrub/emc.glopara/ICS/2017073118/gdas.t18z.abias
lrwxrwxrwx 1 Joe.Schmo g01 76 Dec  5 17:37 gdas.t18z.abias_air -> /gpfs/hps3/emc/global/noscrub/emc.glopara/ICS/2017073118/gdas.t18z.abias_air
lrwxrwxrwx 1 Joe.Schmo g01 75 Dec  5 17:37 gdas.t18z.abias_pc -> /gpfs/hps3/emc/global/noscrub/emc.glopara/ICS/2017073118/gdas.t18z.abias_pc
lrwxrwxrwx 1 Joe.Schmo g01 74 Dec  5 17:37 gdas.t18z.radstat -> /gpfs/hps3/emc/global/noscrub/emc.glopara/ICS/2017073118/gdas.t18z.radstat
```

Set up experiment - part 3

Run workflow setup script from ush/rocoto folder:

Cycled experiment:

> ./setup_workflow.py --expdir $EXPDIR/$PSLOT/

Forecast-only experiment:

> ./setup_workflow_fcstonly.py --expdir $EXPDIR/$PSLOT/

```
SURGE-slogin1 > ./setup_workflow.py --expdir /gpfs/hps3/emc/global/noscrub/Joe.Schmo/para_gfs/fv3demo
sourcing config.prep
sourcing config.anal
sourcing config.fcst
...
sourcing config.epos
sourcing config.earc
```

Set up experiment - part 3

- You will now have a rocoto xml jobcard in your EXPDIR ($PSLOT.xml)

```
SURGE-slogin1 > pwd
/gpfs/hps3/emc/global/noscrub/Joe.Schmo/para_gfs/fv3demo
SURGE-slogin1 > ls
config.anal   config.base.default  config.efcs          config.eupd  config.fv3ic  config.prep       config.vrfy
config.arch   config.earc          config.eobs          config.fcst  config.getic  config.prepbufr
              fv3demo.crontab
config.base   config.ecen          config.epos          config.fv3   config.post   config.resources
              fv3demo.xml
```

- You now also have a crontab file generated for you based on your experiment setup. More on this in a later slide...

Set up experiment - part 3 - common errors

- Message about crontab file not being generated when running setup_workflow*.py script.
  - Cause: If you do not have the rocoto module loaded when you run setup_workflow*.py it will not be able to generate the crontab file for you.
  - Fix: Load rocoto module and run setup_workflow*.py script again.

Start the run

- Make sure the rocoto module is loaded: module load rocoto
- Start your run from within your EXPDIR:

  > rocotorun -d $PSLOT.db -w $PSLOT.xml

- The first jobs of the run should now be queued and starting (depending on machine traffic). How exciting!

- You'll now have a "logs" folder in both your COMROT and EXPDIR. More on this in next slides...

Log folders and files

- There is now a ‘log’ folder in both your EXPDIR and COMROT folders.
- The EXPDIR log folder contains workflow log files:

```
SURGE-slogin1 > pwd
/gpfs/hps3/emc/global/noscrub/Joe.Schmo/para_gfs/fv3demo
SURGE-slogin2 > ls -l logs/
total 128
-rw-r--r-- 1 Joe.Schmo global 536 Dec  5 17:43 2017073118.log
```

- The COMROT log folder contains log files for each job of the run (previously known as dayfiles). See next slide for more...

Log folders and files

COMROT log files are divided into folders for each cycle:

```
SURGE-slogin1 > pwd
/gpfs/hps2/ptmp/Joe.Schmo/fv3demo

SURGE-slogin2 > ls -l logs/
total 0
drwxr-xr-x 2 Joe.Schmo g01 512 Dec  5 17:43 2017073118

SURGE-slogin2 > ls -l logs/2017073118/
total 12288
-rw-r--r-- 1 Joe.Schmo g01 338032 Dec  5 17:45 gdasefcs01.log
-rw-r--r-- 1 Joe.Schmo g01 338043 Dec  5 17:45 gdasefcs02.log
-rw-r--r-- 1 Joe.Schmo g01 388420 Dec  5 17:45 gdasfcst.log
```

Setup experiment cron job

After running setup_workflow.py you should also have a generated crontab file in your EXPDIR.

Set this crontab:

> **> crontab $PSLOT.crontab**
>
> Warning, this will overwrite your existing crontab on your login node. You can alternatively open the crontab file and copy the actual cron command to add to your existing crontab file.

Check it was set correctly:

> **> crontab -l**

Crontab uses following format:

```
*/5 * * * * /path/to/rocotorun -w /path/to/workflow/definition/file -d /path/to/workflow/database/file
```

Optional: Setup symlink to viewer utility for ease of use

- Within your EXPDIR set up a symbolic link to the viewer:

  > ln -s $BASE_SUPER/ush/rocoto/rocoto_viewer.py rocoto_viewer.py

  ...where $BASE_SUPER is the top of the copy of the fv3gfs clone you are using.

- You'll now have a linked copy of rocoto_viewer.py in your EXPDIR.

- Alternatively you could place a copy in your ~/bin folder for use anywhere on the machine. It's up to you!

Monitor experiment using viewer

- Issue the following command while in EXPDIR*:

  > ./rocoto_viewer.py -d $PSLOT.db -w $PSLOT.xml

- Viewer will yell at you if the window is not wide enough to display all of the experiment information columns.

* Note: the viewer requires the full path to the database and xml files if you are not in your EXPDIR when you invoke it. Similarly, the generated crontab file includes that full paths for the rocotorun command, as well as the experiment database and xml files.

The Rocoto Viewer - crash course

Cycle in first column:

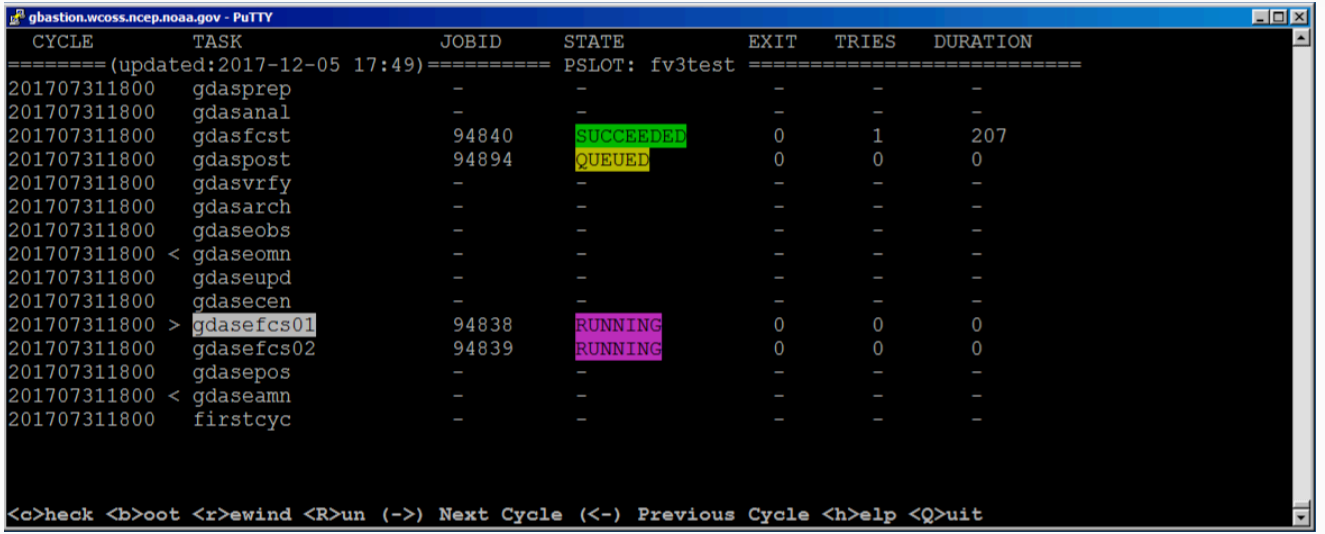
YYYYMMDDCCmm

YYYY=year, MM=month, DD=day
CC=cycle, mm=minute

Task in second column:

< = group/meta-task indicator
Click "**x**" while selecting group task to expand/collapse

States:
QUEUED, RUNNING,
SUCCEEDED, FAILED, DEAD



Example screenshot of the viewer during the first half cycle shortly after the gdasfcst job completed. The gdasefmn metatask is expanded ("x" key) to show the two gdasefcs## subtasks.

Viewer accepts mouse and keyboard inputs. Click "h" for help menu and more options.

c = get information on selected job
r = rewind (rerun) selected job cycle
R = run rocotorun

-> = right arrow key, advance viewer forward to next cycle
<- = left arrow key, advance viewer backward to previous

Q = quit/exit viewer

The Rocoto Viewer - Advanced features

- Select multiple tasks at once (i.e. for rewinding)
  - Click “Enter” on a task to select it, click on other tasks or use the up/down arrows to move to other tasks and click “Enter” to select them as well.
  - When you next choose “r” for rewinding the pop-up window will now ask if you are sure you want to rewind all those selected tasks.
- <r> - Rewind entire group or cycle
  - Group - While group/metatask is collapsed (<) click “r” to rewind whole group/metatask.
  - Cycle - Use up arrow to move selector up past the first task until the entire left column is highlighted. Click “r” and the entire cycle will be rewound.
- <b> - Boot
  - rocotoboot command, will force task to run, ignores dependencies.
  - Click “b” while selecting a task or tasks. Choose “Y” similar to rewind prompt and selected tasks will be booted.
  - Use only if absolutely necessary!

Full documentation of rocoto options can be viewed here:
https://github.com/christopherwharrop/rocoto/wiki/Documentation

Rocoto commands of use

Check the status of the workflow:

rocotostat -w /path/to/workflow/xml/file -d /path/to/workflow/database/file [-c YYYYMMDDHHMM,[YYYYMMDDHHMM,...]] [-t taskname,[taskname,...]] [-s] [-T]

rocotocheck -w /path/to/workflow/xml/file -d /path/to/workflow/database/file -c YYYYMMDDHHMM -t taskname

Force a task to run (ignores dependencies):

rocotoboot -w /path/to/workflow/xml/file -d /path/to/workflow/database/file -c YYYYMMDDHHMM -t taskname

Rerun task(s):

rocotorewind -w /path/to/workflow/xml/file -d /path/to/workflow/database/file -c YYYYMMDDHHMM -t taskname

Several dates and task names may be specified in the same command by adding more -c and -t options. However, lists are not allowed.

Contents of COMROT and file naming convention

Example COMROT folders after a 2.5 cycle run:

SURGE-slogin2 > ls
enkf.gdas.20170731 enkf.gdas.20170801 gdas.20170731 gdas.20170801 gfs.20170801 logs vrfyarch

COMROT structure:

- The COMROT is no longer one flat folder holding everything (hooray!). It is now more similar to how NCO structures their /com folder in production.
- It is broken down by CDUMP (gdas, gfs, enkf.gdas), date (YYYYMMDD), and cycle (CC) directories.
- Log files (formerly known as dayfiles) are now held in a separate ‘logs’ folder at top of COMROT.
- Files for verification and archival are now held in a separate ‘vrfyarch’ folder at top of COMROT.

File naming convention:

- A production-like naming convention is now being employed ($CDUMP.t$CCz.$FILETYPE).
- The parallel naming convention ($FILETYPE.$CDUMP.$CDATE) is being retired to meet EE2 standards.
- All utilities scripts (i.e. chgres) will use the prod-like naming convention so you will need to rename prior parallel output).

Contents of COMROT (gdas.20170731/18)

```
SURGE-slogin2 > ls -l gdas.20170731/18/
total 7361928
lrwxrwxrwx 1 Joe.Schmo g01       75 Dec  5 17:37 INPUT -> /gpfs/hps3/emc/global/noscrub/
emc.glopara/ICS/2017073118/C384/control/INPUT
drwxr-xr-x 2 Joe.Schmo g01     4096 Dec  5 17:47 RESTART
lrwxrwxrwx 1 Joe.Schmo g01       72 Dec  5 17:37 gdas.t18z.abias -> /gpfs/hps3/emc/global/
noscrub/emc.glopara/ICS/2017073118/gdas.t18z.abias
lrwxrwxrwx 1 Joe.Schmo g01       76 Dec  5 17:37 gdas.t18z.abias_air -> /gpfs/hps3/emc/global/
noscrub/emc.glopara/ICS/2017073118/gdas.t18z.abias_air
lrwxrwxrwx 1 Joe.Schmo g01       75 Dec  5 17:37 gdas.t18z.abias_pc -> /gpfs/hps3/emc/global/
noscrub/emc.glopara/ICS/2017073118/gdas.t18z.abias_pc
-rw-r--r-- 1 Joe.Schmo g01 608722964 Dec  5 17:44 gdas.t18z.atmf000.nemsio
-rw-r--r-- 1 Joe.Schmo g01 608722964 Dec  5 17:45 gdas.t18z.atmf003.nemsio
-rw-r--r-- 1 Joe.Schmo g01 608722964 Dec  5 17:45 gdas.t18z.atmf006.nemsio
-rw-r--r-- 1 Joe.Schmo g01 608722964 Dec  5 17:46 gdas.t18z.atmf009.nemsio
-rw-r--r-- 1 Joe.Schmo g01 181244765 Dec  5 17:49 gdas.t18z.pgrb2.0p25.f000
-rw-r--r-- 1 Joe.Schmo g01     16214 Dec  5 17:49 gdas.t18z.pgrb2.0p25.f000.idx
-rw-r--r-- 1 Joe.Schmo g01 227262085 Dec  5 17:51 gdas.t18z.pgrb2.0p25.f003
-rw-r--r-- 1 Joe.Schmo g01     24672 Dec  5 17:51 gdas.t18z.pgrb2.0p25.f003.idx
-rw-r--r-- 1 Joe.Schmo g01 230895899 Dec  5 17:53 gdas.t18z.pgrb2.0p25.f006
-rw-r--r-- 1 Joe.Schmo g01     24673 Dec  5 17:53 gdas.t18z.pgrb2.0p25.f006.idx
-rw-r--r-- 1 Joe.Schmo g01 232746569 Dec  5 17:54 gdas.t18z.pgrb2.0p25.f009
-rw-r--r-- 1 Joe.Schmo g01     24677 Dec  5 17:54 gdas.t18z.pgrb2.0p25.f009.idx
-rw-r--r-- 1 Joe.Schmo g01  58158730 Dec  5 17:49 gdas.t18z.pgrb2.0p50.f000
-rw-r--r-- 1 Joe.Schmo g01     16010 Dec  5 17:49 gdas.t18z.pgrb2.0p50.f000.idx
-rw-r--r-- 1 Joe.Schmo g01  72429410 Dec  5 17:51 gdas.t18z.pgrb2.0p50.f003
-rw-r--r-- 1 Joe.Schmo g01     24374 Dec  5 17:51 gdas.t18z.pgrb2.0p50.f003.idx
-rw-r--r-- 1 Joe.Schmo g01  73510837 Dec  5 17:53 gdas.t18z.pgrb2.0p50.f006
-rw-r--r-- 1 Joe.Schmo g01     24375 Dec  5 17:53 gdas.t18z.pgrb2.0p50.f006.idx
-rw-r--r-- 1 Joe.Schmo g01  73999248 Dec  5 17:54 gdas.t18z.pgrb2.0p50.f009
-rw-r--r-- 1 Joe.Schmo g01     24378 Dec  5 17:54 gdas.t18z.pgrb2.0p50.f009.idx
-rw-r--r-- 1 Joe.Schmo g01  18208653 Dec  5 17:49 gdas.t18z.pgrb2.1p00.f000
-rw-r--r-- 1 Joe.Schmo g01     15868 Dec  5 17:49 gdas.t18z.pgrb2.1p00.f000.idx
-rw-r--r-- 1 Joe.Schmo g01  22431753 Dec  5 17:51 gdas.t18z.pgrb2.1p00.f003
-rw-r--r-- 1 Joe.Schmo g01     24230 Dec  5 17:51 gdas.t18z.pgrb2.1p00.f003.idx
-rw-r--r-- 1 Joe.Schmo g01  22706872 Dec  5 17:53 gdas.t18z.pgrb2.1p00.f006
-rw-r--r-- 1 Joe.Schmo g01     24232 Dec  5 17:53 gdas.t18z.pgrb2.1p00.f006.idx
-rw-r--r-- 1 Joe.Schmo g01  22824981 Dec  5 17:54 gdas.t18z.pgrb2.1p00.f009
-rw-r--r-- 1 Joe.Schmo g01     24233 Dec  5 17:54 gdas.t18z.pgrb2.1p00.f009.idx
-rw-r--r-- 1 Joe.Schmo g01  34032854 Dec  5 17:49 gdas.t18z.pgrbf00
-rw-r--r-- 1 Joe.Schmo g01  40667146 Dec  5 17:51 gdas.t18z.pgrbf03
-rw-r--r-- 1 Joe.Schmo g01  40652330 Dec  5 17:53 gdas.t18z.pgrbf06
-rw-r--r-- 1 Joe.Schmo g01  40815788 Dec  5 17:54 gdas.t18z.pgrbf09
-rw-r--r-- 1 Joe.Schmo g01 547623484 Dec  5 17:49 gdas.t18z.pgrbq00
-rw-r--r-- 1 Joe.Schmo g01 655353800 Dec  5 17:51 gdas.t18z.pgrbq03
-rw-r--r-- 1 Joe.Schmo g01 655955746 Dec  5 17:53 gdas.t18z.pgrbq06
-rw-r--r-- 1 Joe.Schmo g01 657337304 Dec  5 17:54 gdas.t18z.pgrbq09
lrwxrwxrwx 1 Joe.Schmo g01       74 Dec  5 17:37 gdas.t18z.radstat -> /gpfs/hps3/emc/global/
noscrub/emc.glopara/ICS/2017073118/gdas.t18z.radstat
-rw-r--r-- 1 Joe.Schmo g01 138027404 Dec  5 17:44 gdas.t18z.sfcf000.nemsio
-rw-r--r-- 1 Joe.Schmo g01 138027404 Dec  5 17:45 gdas.t18z.sfcf003.nemsio
-rw-r--r-- 1 Joe.Schmo g01 138027404 Dec  5 17:45 gdas.t18z.sfcf006.nemsio
-rw-r--r-- 1 Joe.Schmo g01 138027404 Dec  5 17:46 gdas.t18z.sfcf009.nemsio
-rw-r--r-- 1 Joe.Schmo g01  14502858 Dec  5 17:48 gdas.t18z.sfluxgrbf000.grib2
-rw-r--r-- 1 Joe.Schmo g01      6150 Dec  5 17:48 gdas.t18z.sfluxgrbf000.grib2.idx
-rw-r--r-- 1 Joe.Schmo g01  15273279 Dec  5 17:50 gdas.t18z.sfluxgrbf003.grib2
-rw-r--r-- 1 Joe.Schmo g01      6796 Dec  5 17:50 gdas.t18z.sfluxgrbf003.grib2.idx
-rw-r--r-- 1 Joe.Schmo g01  15862633 Dec  5 17:52 gdas.t18z.sfluxgrbf006.grib2
-rw-r--r-- 1 Joe.Schmo g01      6801 Dec  5 17:52 gdas.t18z.sfluxgrbf006.grib2.idx
-rw-r--r-- 1 Joe.Schmo g01  15763687 Dec  5 17:53 gdas.t18z.sfluxgrbf009.grib2
-rw-r--r-- 1 Joe.Schmo g01      6801 Dec  5 17:53 gdas.t18z.sfluxgrbf009.grib2.idx
-rw-r--r-- 1 Joe.Schmo g01 130253958 Dec  5 17:48 pgrbm000.gdas.2017073118.grib2
-rw-r--r-- 1 Joe.Schmo g01    153328 Dec  5 17:48 pgrbm000.gdas.2017073118.grib2.idx
-rw-r--r-- 1 Joe.Schmo g01 146767789 Dec  5 17:50 pgrbm003.gdas.2017073118.grib2
-rw-r--r-- 1 Joe.Schmo g01    171890 Dec  5 17:50 pgrbm003.gdas.2017073118.grib2.idx
-rw-r--r-- 1 Joe.Schmo g01 148899436 Dec  5 17:51 pgrbm006.gdas.2017073118.grib2
-rw-r--r-- 1 Joe.Schmo g01    171946 Dec  5 17:51 pgrbm006.gdas.2017073118.grib2.idx
-rw-r--r-- 1 Joe.Schmo g01 149467229 Dec  5 17:53 pgrbm009.gdas.2017073118.grib2
-rw-r--r-- 1 Joe.Schmo g01    171918 Dec  5 17:53 pgrbm009.gdas.2017073118.grib2.idx
```

Build workflow documentation using doxygen

Within the /docs folder:

> ./compile

You will now have new folders called “html” and “latex”:

```
SURGE-slogin2 > ls -l
total 408
-rwxr-xr-x 1 Joe.Schmo global    641 Aug 21 15:26 compile
-rw-r--r-- 1 Joe.Schmo global 102968 Aug 21 15:26 doxyfile
drwxr-sr-x 3 Joe.Schmo global   8192 Sep 19 18:15 html
drwxr-sr-x 2 Joe.Schmo global   4096 Sep 19 18:15 latex
-rw-r--r-- 1 Joe.Schmo global   8155 Aug 21 15:26 mainpage.h
```

Appendix

Following slides contain additional information, including document history.

Document information - history of changes

Version 3.4 changes (March 1st, 2018)

- Updated instructions for enhanced build system. The build_all.sh script now supports partial build capability; see sorc/fv3gfs_build.cfg config file for control of partial build. Modify before running build_all.sh if doing partial build. Machine argument no longer required as argument to build_all.sh; new build scripts autodetect machine.

Version 3.3 changes (February 8th, 2018)

- Updates for flattened workflow structure.

Version 3.2 changes (January 17th, 2018)

- Add information on additional setup script options.
- Added link to Ken Sperow VLab presentation.
- Added slide for status of git master.
- Added slide on making your own ICs.
- Added POCs for post/UPP project.
- Small textual fixes and colorized slides for easier navigation between sections.

Document information - history of changes

Version 3.1 changes (December 13th, 2017)

- Add information on rocoto commands for use outside of viewer.

Version 3.0 changes (December 12th, 2017):

- Add information and instructions for forecast-only experiments.
- Small textual fixes.

Version 2.0 changes (December 5th, 2017):

- Updated “Important pages and documentation”. Added point-of-contact information for sub-modules.
- Updated to use fv3gfs git master instead of merge-update branch.
- Updated access and clone instructions.
- New supported resolution initial condition information.
- Removed “--machine” from setup_expt.py instructions. No longer needed in command.
- Updated sample screenshot of Rocoto viewer.