

# Doxygen Documentation for the UPP

By: Gillian Petro

A dark blue diagonal gradient bar that starts from the bottom left corner and extends towards the top right corner, covering the lower half of the page.

# Background

- EPIC (and, previously, the DTC) has been working to fully document UPP code.
  - Thank you to Tracy Hertnecky and Kate Fossell from DTC for a smooth documentation transition to EPIC.
  - Thank you to Ed Hartnett at EMC for his ongoing assistance.
- **Goal:** Automate documentation testing and enforce standards around documentation.
  - Ultimately, this will make working with code easier/faster!

# Progress

- EPIC is documenting existing code as thoroughly as possible.
- Issue #392: <https://github.com/NOAA-EMC/UPP/issues/392>
- Pull Requests:
  - PR [#630](#) & PR [#659](#) (CALGUST.f and CALHEL.f)
  - PR [#663](#) & PR [#665](#) (GFSPPOSTSIG.f)
  - PR [#681](#) & PR [#689](#) (MSFPS.f, NGMFLD.f, OTLFT.f, OTLIFT.f, and ZENSUN.f)
  - PR [#696](#) & PR [#697](#) (updates to 23 files)
  - PR [#698](#) & PR [#701](#) (updates to 12 files)
  - PR [#737](#) & PR [#757](#) (updates to 28 files)

# TO DO:

- Test/approve [PR #932](#), which builds documentation with the CI
  - Will not fail for documentation warnings
- Document modules
  - Underway
- Turn on WARN\_AS\_ERROR flag
  - Documentation must be provided for newly added variables, functions, subroutines, and files

# What are developers responsible for?

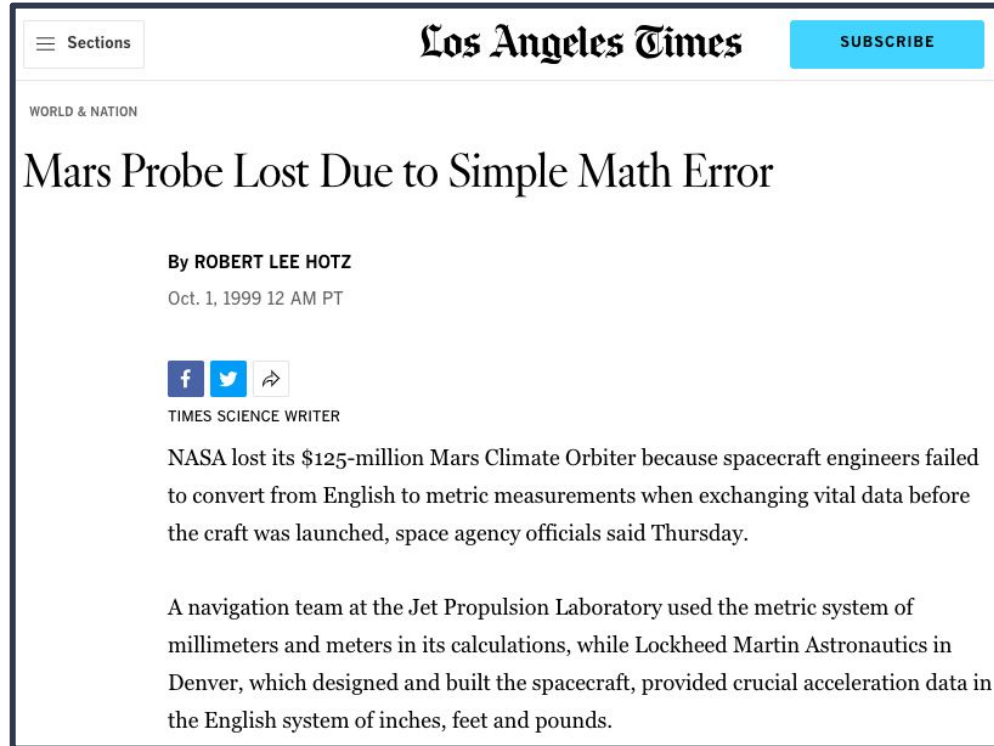
- *You are only responsible for documenting your own code contributions! :)*
- NOT responsible for other people's code

# Why document code?

- **Bad (or nonexistent) documentation costs time and money!**
- Good documentation *saves time* by:
  - Clarifying what the code is doing → no need to search the code or email your colleague to figure it out!
  - Facilitating knowledge transfer for:
    - New team members
    - Community developers
    - User support staff & CMs
  - Improving adoption of UPP and UFS software by community developers
    - Expanding developer community → shares the work!
- More time && more developers == more science!

# Why document code?

- Bad documentation **can cost millions!**
  - Example courtesy of Ed Hartnett
- We are not the only ones who use our documentation.
  - Users come from academia, government/military, industry
- Excellence matters in science AND in documentation of the science!



The screenshot shows the top portion of a news article from the Los Angeles Times. At the top right is a blue 'SUBSCRIBE' button. Below the newspaper name is the section 'WORLD & NATION'. The article title is 'Mars Probe Lost Due to Simple Math Error'. The author is 'By ROBERT LEE HOTZ' and the date is 'Oct. 1, 1999 12 AM PT'. There are social media sharing icons for Facebook, Twitter, and a general share icon. Below the icons is the text 'TIMES SCIENCE WRITER'. The main text of the article begins with 'NASA lost its \$125-million Mars Climate Orbiter because spacecraft engineers failed to convert from English to metric measurements when exchanging vital data before the craft was launched, space agency officials said Thursday.'

Sections

Los Angeles Times

SUBSCRIBE

WORLD & NATION

## Mars Probe Lost Due to Simple Math Error

By ROBERT LEE HOTZ

Oct. 1, 1999 12 AM PT

[f](#) [t](#) [↗](#)

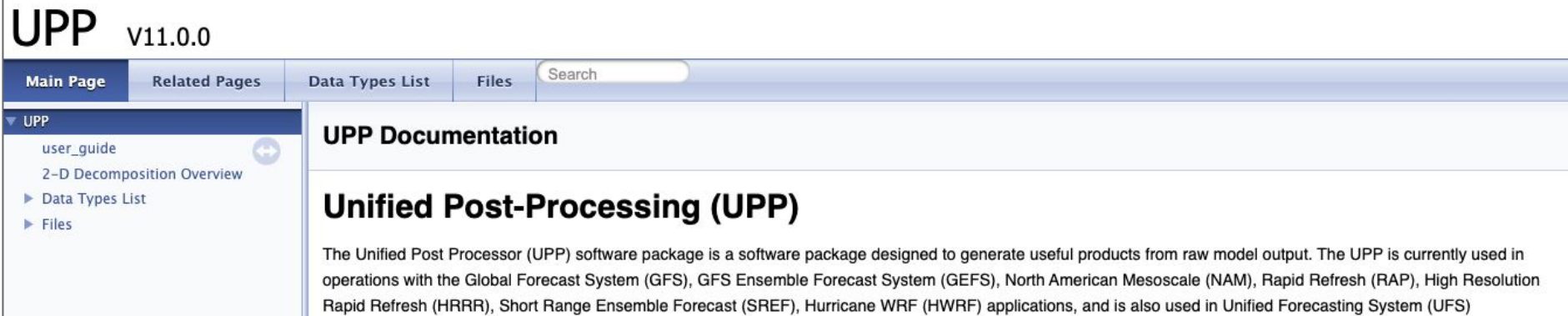
TIMES SCIENCE WRITER

NASA lost its \$125-million Mars Climate Orbiter because spacecraft engineers failed to convert from English to metric measurements when exchanging vital data before the craft was launched, space agency officials said Thursday.

A navigation team at the Jet Propulsion Laboratory used the metric system of millimeters and meters in its calculations, while Lockheed Martin Astronautics in Denver, which designed and built the spacecraft, provided crucial acceleration data in the English system of inches, feet and pounds.

# What is Doxygen?

- A widely used documentation generator for scientific software development.
  - Pulls content directly from source code files
  - Renders it in human-friendly/human readable form
- Used in projects like HDF5, netCDF → and NCEPLIBS!
- Handles Fortran code well



**UPP** V11.0.0

Main Page | Related Pages | Data Types List | Files | Search

UPP

- user\_guide
- 2-D Decomposition Overview
- Data Types List
- Files

## UPP Documentation

### Unified Post-Processing (UPP)

The Unified Post Processor (UPP) software package is a software package designed to generate useful products from raw model output. The UPP is currently used in operations with the Global Forecast System (GFS), GFS Ensemble Forecast System (GEFS), North American Mesoscale (NAM), Rapid Refresh (RAP), High Resolution Rapid Refresh (HRRR), Short Range Ensemble Forecast (SREF), Hurricane WRF (HWRF) applications, and is also used in Unified Forecasting System (UFS)



# Doxyfile.in

- Configures Doxygen documentation
- Located at: `UPP/doc/Doxyfile.in`
- Running `doxygen Doxyfile.in` generates warnings for undocumented variables, functions, subroutines when `WARN_AS_ERROR = YES`
- Build fails

```
WARN_NO_PARAMDOC      = YES
```

```
# If the WARN_AS_ERROR tag is set to YES then doxygen will immediately stop when  
# a warning is encountered. If the WARN_AS_ERROR tag is set to FAIL_ON_WARNINGS  
# then doxygen will continue running as if WARN_AS_ERROR tag is set to NO, but  
# at the end of the doxygen process doxygen will return with a non-zero status.  
# Possible values are: NO, YES and FAIL_ON_WARNINGS.  
# The default value is: NO.
```

```
WARN_AS_ERROR         = NO
```

# Routines & Subroutines: Code

- Describe each subroutine
- Describe each variable
- Add yourself to History Log

```
!> @file
!> @brief Subroutine that computes storm relative helicity.
!
!> This routine computes estimated storm motion and storm-relative
!> environmental helicity. (Davies-Jones et al 1990) the algorithm
!> processed as follows.
!> 2022-10-07 | Tracy Hertneky | Add left mover for storm motion in SH
!>
!> @author Michael Baldwin W/NP2 @date 1994-08-22
SUBROUTINE CALHEL(DEPTH,UST,VST,HELI,USHR1,VSHR1,USHR6,VSHR6)
!> @param[in] DEPTH Depth in meters over which helicity should be computed; allows one to disti
!> @param[out] UST Estimated U Component (m/s) Of Storm motion.
!> @param[out] VST Estimated V Component (m/s) Of Storm motion.
!> @param[out] HELI Storm-relative helicity (m**2/s**2).
!> @param[out] USHR1 U Component (m/s) Of 0-1 km shear.
!> @param[out] VSHR1 V Component (m/s) Of 0-1 km shear.
!> @param[out] USHR6 U Component (m/s) Of 0-0.5 to 5.5-6.0 km shear.
!> @param[out] VSHR6 V Component (m/s) Of 0-0.5 to 5.5-6.0 km shear.
```

# Routines & Subroutines

!> **@file** (tells Doxygen to check for documentation)

!> **@brief** Describe file or subroutine here

!> **@param** *name Description units*

!> @param[in]

!> @param[out]

!> @param[inout]

!> **@return** *varname Description* (often, the return value is the same as the subroutine name)

- For Fortran, "!>" or "!<" starts a comment and "!!" or "!!>" can be used to continue a one line comment into a multi-line comment.

# Routines & Subroutines (cont'd)

- Find subroutine/function “signature” to see what variables need to be documented.

```
SUBROUTINE CALHEL (DEPTH, UST, VST, HELI, USHR1, VSHR1, USHR6, VSHR6)
```

- Find variable declarations to determine data type and in/out/inout type.

```
70     real, intent(in)                :: DEPTH(2)
71     REAL, dimension(ista_2l:iend_2u, jsta_2l:jend_2u), intent(out) :: UST, VST
72     REAL, dimension(ista_2l:iend_2u, jsta_2l:jend_2u, 2), intent(out) :: HELI
73     !
74     real, dimension(ista_2l:iend_2u, jsta_2l:jend_2u) :: HTSFC, UST6, VST6, UST5, VST5, &
75     UST1, VST1, USHR1, VSHR1, &
76     USHR6, VSHR6, U1, V1, U2, V2, &
```

# Routines & Subroutines cont'd

Example: `sorc/ncep_post.fd/CALHEL.f`

```
SUBROUTINE CALHEL (DEPTH, UST, VST, HELI, USHR1, VSHR1, USHR6, VSHR6)
```

!> @file

!> @brief Subroutine that computes storm relative helicity.

...

!> @param[in] DEPTH Depth in meters over which helicity should be computed; allows one to distinguish 0-3 km and 0-1 km values.

!> @param[out] UST Estimated U Component (m/s) Of Storm motion.

!> @param[out] VST Estimated V Component (m/s) Of Storm motion.

!> @param[out] HELI Storm-relative helicity ( $m^{**2}/s^{**2}$ ).

!> @param[out] USHR1 U Component (m/s) Of 0-1 km shear.

!> @param[out] VSHR1 V Component (m/s) Of 0-1 km shear.

...

# Routines & Subroutines: Authors & History

!> **@author** author name

- Add your change to the History Log

!> ### Program History Log

!> Date | Programmer | Comments

!> ----|-----|-----

!> 2019-09-24 | Y Mao | Rewritten from MISCLN.f

!> 2020-05-20 | J Meng | CALRH unification with NAM scheme

!> 2020-11-10 | J Meng | Use UPP\_PHYSICS Module

!> 2021-03-11 | B Cui | Change local arrays to dimension (im,jsta:jend)

!> 2021-10-14 | J MENG | 2D DECOMPOSITION

!> **YYYY-MM-DD | F Lastname | Description of additions**

# Routines & Subroutines: Rendering

[https://noaa-emc.github.io/UPP/CALHEL\\_8f.html](https://noaa-emc.github.io/UPP/CALHEL_8f.html)

## Functions/Subroutines

subroutine **calhel** (DEPTH, UST, VST, HELI, USHR1, VSHR1, USHR6, VSHR6)

This routine computes estimated storm motion and storm-relative environmental helicity. [More...](#)

## Function/Subroutine Documentation

```
subroutine calhel ( real, dimension(2), intent(in)           DEPTH,
                  real, dimension(ista_2l:iend_2u,jsta_2l:jend_2u), intent(out) UST,
                  real, dimension(ista_2l:iend_2u,jsta_2l:jend_2u), intent(out) VST,
                  real, dimension(ista_2l:iend_2u,jsta_2l:jend_2u,2), intent(out) HELI,
                  real, dimension(ista_2l:iend_2u,jsta_2l:jend_2u)           USHR1,
                  real, dimension(ista_2l:iend_2u,jsta_2l:jend_2u)           VSHR1,
                  real, dimension(ista_2l:iend_2u,jsta_2l:jend_2u)           USHR6,
                  real, dimension(ista_2l:iend_2u,jsta_2l:jend_2u)           VSHR6
                  )
```

## Parameters

- [in] **DEPTH** Depth in meters over which helicity should be computed; allows one to distinguish 0-3 km and 0-1 km values.
- [out] **UST** Estimated U Component (m/s) Of Storm motion.
- [out] **VST** Estimated V Component (m/s) Of Storm motion.
- [out] **HELI** Storm-relative helicity ( $m^{**2}/s^{**2}$ ).
- [out] **USHR1** U Component (m/s) Of 0-1 km shear.
- [out] **VSHR1** V Component (m/s) Of 0-1 km shear.
- [out] **USHR6** U Component (m/s) Of 0-0.5 to 5.5-6.0 km shear.
- [out] **VSHR6** V Component (m/s) Of 0-0.5 to 5.5-6.0 km shear.

# Modules

- Not all variables in modules have documentation
- Regardless, add documentation or comments for the variables you add!
- Each variable declared on a single line (see CTLBLK.f for example)

```
!> @file
!> @brief module: CTLBLK sets default parameters that are used throughout the UPP code

integer :: NUM_PROCS      !< The number of MPI ranks available to the post processor.
integer :: ME             !< MPI rank.
integer :: JSTA           !< Start latitude on a task subdomain.
integer :: JEND           !< End latitude on a task subdomain.
integer :: ISTA           !< Start longitude latitude on a task subdomain.
integer :: IEND           !< End longitude on a task subdomain.
integer :: JSTA_M         !< Beginning latitude loop index in subdomain for halo depth 1.
integer :: JEND_M         !< Ending latitude loop index in subdomain for halo depth 1.
```



Questions?



# Creating a New Routine

- Begin file with `!> @file` to signal to Doxygen that it should read the file.
- Add `!> @brief` statement to describe the file.
- Add any additional description (optional)

```
1    !> @file
2    !> @brief NEWROUTINE computes the area of various shapes.
3    !>
4    !> Additional description here *if desired*
5    !>
```

# Creating a New Routine (cont'd)

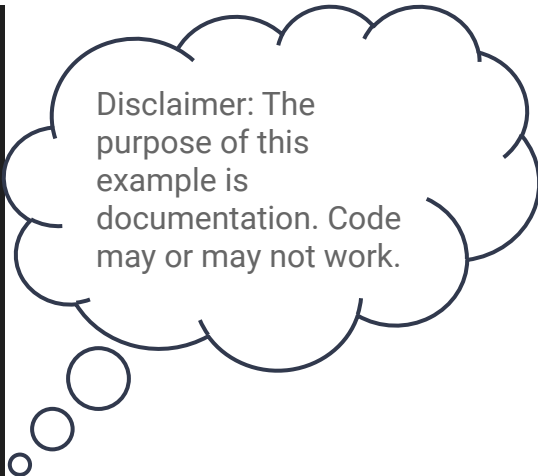
- Add history log
- Add **@author** and **@date**
- Add separator line to visually separate header info from code (not required but helpful)

```
6  !> ### Program history log:
7  !> Date | Programmer | Comments
8  !> ----|-----|-----
9  !> 2024-04-22 | Gillian Petro | Initial code w/area of circle and rectangle
10 !>
11 !> @author Gillian Petro @date 2024-04-22
12 !-----
```

# Creating a New Routine (cont'd)

- Describe purpose of subroutine using **@brief**
- Describe parameters using **@param** *name Description units*

```
13  !> @brief compute_area_rectangle() computes area of a rectangle.
14  !> @param[in] length Length of rectangle in meters.
15  !> @param[in] width Width of rectangle in meters.
16  ✓ !> @param[out] area_rectangle Area of rectangle in m^2.
17  SUBROUTINE compute_area_rectangle(length,width,area_rectangle)
18
19  IMPLICIT NONE
20
21  real,intent(in)      :: length,width
22  real,intent(out)    :: area_rectangle
23
24  area_rectangle = length * width
25
26  END SUBROUTINE area_rectangle
```



Disclaimer: The purpose of this example is documentation. Code may or may not work.

# Best Practices

- Use descriptive variable and function names → makes code easier to read and understand
  - For example, *compute\_area\_rectangle()* and *compute\_area\_circle()* are much clearer than *area1()* and *area2()*.
  - *radius* is clearer than *r*
  - *length* and *width* are clearer than *l* or *w*

# Generate Documentation Locally

- In `UPP/doc/Doxyfile.in` change `@abs_top_srcdir@` and `@config_srcdir@` in `INPUT` and `USE_MDFILE_AS_MAINPAGE` to the path to your local UPP clone:
  - `INPUT` = `@abs_top_srcdir@/doc/user_guide.md \`  
`@abs_top_srcdir@/doc/2D-decomp.md \`  
`@abs_top_srcdir@/sorc/ncep_post.fd \`  
`@config_srcdir@`
  - `USE_MDFILE_AS_MAINPAGE` =  
`@abs_top_srcdir@/docs/sp_user_guide.md`
- For me, `@abs_top_srcdir@` and `@config_srcdir@` change to `/work/noaa/epic/gpetro/UPP` on Orion or Hercules.

# Generate Documentation Locally (cont'd)

- In UPP/doc run:
  - `doxygen Doxyfile.in`
  - If all goes well, there will be no warnings with the “Generating...” message. →

```
Generating docs for file MPI_LAST.f...
Generating docs for file MSFPS.f...
Generating docs for file native_endianness.f...
Generating docs for file NEWROUTINE.f...
Generating docs for file NGMFLD.f...
Generating docs for file NGMSLP.f...
Generating docs for file OTLFT.f...
```

- This will raise warnings if there is a problem:

```
Generating docs for file NEWROUTINE.f...
/Users/gillianpetro/UPP/sorc/ncep_post.fd/NEWROUTINE.f:14: warning: The following p
arameter of compute_area_rectangle(real, intent(in) length, real, intent(in) width,
real, intent(out) area_rectangle) is not documented:
parameter 'width'
```

# HTML Version

- Check UPP/doc/html for the html files
- Do NOT add html files to your PR!
  - They are placed in the *gh-pages* branch later in a second PR by a CM.

The screenshot shows the UPP v11.0.0 documentation website. The top navigation bar includes 'Main Page', 'Related Pages', 'Modules', 'Data Types', and 'Files'. A sidebar on the left lists various modules, with 'NEWROUTINE.f' highlighted. The main content area displays the definition of the 'compute\_area\_rectangle()' subroutine, including its parameters and a brief description.

UPP v11.0.0

Main Page Related Pages Modules Data Types Files

MDLFLD.f  
MICROINIT.F  
▶ MISCLN.f  
MIXLEN.f  
MPI\_FIRST.f  
▶ MPI\_LAST.f  
▶ MSFPS.f  
▶ native\_endianness.f  
▶ NEWROUTINE.f  
▶ NGMFLD.f  
▶ NGMSLP.f  
▶ OTLFT.f  
▶ OTLIFT.f  
▶ PARA\_RANGE.f  
PARAMR.f  
▶ params.F  
▶ physcons.f  
▶ PMICRPH.f  
POLFAVG.f

◆ compute\_area\_rectangle()

```
subroutine compute_area_rectangle ( real, intent(in) length,  
                                  real, intent(in) width,  
                                  real, intent(out) area_rectangle  
)
```

compute\_area\_rectangle() computes area of a rectangle.

**Parameters**

- [in] **length** Length of rectangle in meters.
- [in] **width** Width of rectangle in meters.
- [out] **area\_rectangle** Area of rectangle in m<sup>2</sup>.

Definition at line 17 of file NEWROUTINE.f.

sorc ncep\_post.fd NEWROUTINE.f



Questions?



# Will there be a CMake option to change Doxyfile.in and build the documentation?

- This option currently exists in UPP! To enable Doxygen before running CMake:
  - In UPP/CMakeLists.txt, change ENABLE\_DOCS option from OFF to ON.

```
option(ENABLE_DOCS "Enable generation of doxygen-based documentation." OFF)
```

- Run compile\_upp.sh as usual in the tests directory.
    - This will generate a tests/build/doc directory.
- Another option to build *docs only* via CMake:
  - Change tests/compile\_upp.sh script to say `make doc` instead of `make install`.
  - This will only generate the html files from source code and will *not build* the UPP.

# When documenting on WCOS2 or another remote system, what's the cleanest way to view the html?

- Generate the HTMLs in your clone of the UPP by running `doxygen Doxyfile.in` according to the instructions on previous slides.
- On some RDHPCS, it may be possible to securely copy (`scp`) the `html` directory to your local system and open files in the browser on your local system. (See RDHPCS documentation on [Transferring Data](#).)
- When a file transfer is not possible, developers can add the HTMLs to the `gh-pages` branch of their UPP fork and activate GitHub Pages to build documentation from these HTMLs.




# When documenting on WCOS2 or another remote system, what's the cleanest way to view the html? (cont'd)

- ***NOTE:** The HTML files in UPP reside in the gh-pages branch. This keeps source code and compiled code separate but also causes some inconvenience for developers who want to add HTML files to the gh-pages branch in their UPP fork and view them through GitHub Pages. There does not seem to be a “clean” way to do this, but here are a few suggestions:*
- Clone an HTML-specific copy of UPP and set remotes:
  - `git clone -b gh-pages https://github.com/NOAA-EMC/UPP.git UPP-html`
  - `cd UPP-html`
  - `git remote add upstream https://github.com/NOAA-EMC/UPP.git`
  - `git remote set-url origin https://github.com/<your-user-name>/UPP.git`

# When documenting on WCOS2 or another remote system, what's the cleanest way to view the html? (cont'd)

- Optionally, users can create their own branch that tracks *gh-pages*. For example:
  - `git checkout -b text/ghp`
- Copy HTML files from the original UPP clone where you generated them to the UPP-html clone (modify path accordingly):
  - `cp /path/to/<UPP-original>/doc/html/*.html .`
  - `git add -u`
  - `git commit -m "add new htmls"`
  - `git push origin <branch_name>`
    - `<branch_name>` can be *gh-pages* or the name of a branch that tracks *gh-pages* (e.g., `text/ghp`).
- Once the HTML files are in the *gh-pages* branch of your UPP fork, you can activate GitHub pages.

# When documenting on WCOS2 or another remote system, what's the cleanest way to view the html? (cont'd)

- To [activate GitHub pages](#):
  - Navigate to your repository.
  - Click on *Settings* in the repository.  **Settings**
  - Click on *Pages* in the sidebar (under "Code and automation").  **Pages**
  - Under "Build and deployment" → "Source", select Deploy from a branch.
  - Use the branch dropdown menu to select "gh-pages" as the publishing source.
  - Use the folder dropdown menu to select "/doc" as the publishing source.
  - Click "Save".
  - To see your published site, under "GitHub Pages", click "Visit site" or navigate to `https://your-github-username.github.io/UPP`.  **Visit site**

# Resources

- **Doxygen Documentation:** <https://www.doxygen.nl/manual/commands.html>
  - Note that “\” is replaced by “@” in our docs
- **DTC Internal Guidance Document:**  
[https://docs.google.com/document/d/1\\_w2yqLIV6zVU251D62SOPH-lx\\_Xufj6apBHhtrvgz5A/edit](https://docs.google.com/document/d/1_w2yqLIV6zVU251D62SOPH-lx_Xufj6apBHhtrvgz5A/edit)
- **UPP Developer Support page:**  
<https://github.com/NOAA-EMC/UPP/wiki/UPP-Code-Development>
  - Contains info on:
    - Contribution process (including resolving merge conflicts)
    - Testing
    - Doxygen Documentation