

**Standards of Execution for the Aviation Weather Center/Aviation  
Weather Testbed**

## Contents

Standards of Execution for the Transfer of Technology to the Aviation Weather Center.....	2
Deliverable of Scientific Concept or Methodology:.....	3
Definitions:.....	3
Proposals:.....	3
Process.....	3
Product Evaluation.....	3
Deliverable of Web Tools or Products to the Aviation Digital Data Service (ADDS).....	4
Definitions.....	4
Specifications.....	4
Web Visualization.....	4
Deliverable of System code for transfer to AWC:.....	5
Definitions.....	5
Specifications and Documentation.....	5
Deliverable of System code for transfer to NCEP via AWC:.....	6
Definitions.....	6
Specifications and Documentation.....	6
Procedure.....	7

## **Standards of Execution for the Transfer of Technology to the Aviation Weather Center**

- This document defines the standards of execution for any technology transfer to the Aviation Weather Center (AWC) /Aviation Weather Testbed (AWT) from an outside institution, meeting the missions and objectives as outlined by the NOAA Aviation Weather Testbed Charter. These technologies are defined as either a post-processing technology to better enhance aviation forecasting at the AWC/AWT or as aviation specific web tools that may fit within the framework of the AWC/Aviation Digital Data Service. This will cover, not only standards for technology constructed to fit within the framework of our current support level and technology, but also those technologies with which we support on a national level.

## Deliverable of Scientific Concept or Methodology:

### Definitions:

- A scientific concept or methodology may be a new or improved aviation specific variable forecast or analysis that is intended for use by aviation meteorologists and/or flight planners with meteorological knowledge.

### Proposals:

- Any scientific concept or methodology must be submitted to the AWT Director via a short proposal. See the AWT Charter at <http://testbed.aviationweather.gov> for the AWT Director and governance policy. Any proposal should outline the following:
  - What is the intent of your concept or product?  
What scientific purpose is your product addressing and how will it advance forecasting/aviation planning?  
Is this an aviation related concept for use by the AWC or is the intended audience users of the web tools made available on ADDS?
  - What are the computing resources that needed to run your product?  
Is the intent to run at AWC or on a national infrastructure such as WCOSS supercomputer? Serial or Parallel?  
Will this run on current operating systems at AWC?  
Is your output graphical or data, or both?
  - What is the intended support level for your finalized product?  
Is this reliant on regularly scheduled model data or observation data?  
What is the update frequency and how many forecast hours does it produce?
  - What is the operational path intended for your finalized product?

### Process

- Initial Product Description Documentation
  - All initial scientific concepts and methodologies will be passed through the AWT at the AWC  
These may be passed through either by a document explaining the concept or methodology or through a product to be visualized and tested.
  - Do work with AWC developers and forecasters or any potential users and customers to gain feedback before progressing too far down into development
- Data Transfer and Formats
  - If the concept or methodology is presented as a product to be visualized, data (forecast and/or analysis) shall be passed via some form of data transfer (i.e. http, ftp, ldm, etc...)
  - The data shall be in NCEP GRIB2 data format

### Product Evaluation

- Any forecast algorithm, tool, or decision aid must also be subject to the National Weather Service evaluation period for operational transition (<http://www.nws.noaa.gov/directives/sym/pd01001002cur.pdf>)

## Deliverable of Web Tools or Products to the Aviation Digital Data Service (ADDS)

### Definitions

- A deliverable web tool is defined as a product to be placed on ADDS that offers functionality beyond data visualization. See current Flight Path Tool as an example of a web tool (<http://www.aviationweather.gov/adds/ftapplication/>)
- A deliverable of a product for visualization is defined as a post-processed algorithm that will be visualized in a static graphic on the ADDS website for operational use. The algorithm to be transferred must follow the standards as set for deliverable of scientific concept or methodology.

### Specifications

- Tools must meet the following code specifications
  - In order to support mobile platforms, tools must not use client side applications such as Java, but instead using browser based applications such as HTML5 and OpenLayers
  - Tools must utilize open geospatial standards utilizing OpenLayers and MapServer
  - Tools must be scalable (zoom out globally or zoom in regionally)
- Do provide well documented procedures for your tool and any installation directions.
  - Make sure this is continually updated and contains any needed tests
- Keep a version control of any code that is submitted
- If any updates are required post installation, please provide routine update patches
  - Patches must also work with the latest developed version of the tools

### Web Visualization

- Products developed for ADDS static visualizations must utilize the MapPlotlib and Basemap python libraries
  - Other plotting exceptions may include python interfaces for NCL (PyNGL)
- Static images for use on the web must produce a .png or .gif file
- Products developed for ADDS visualization must pass graphics through Safety Risk Management within the FAA before it may become operational ([https://www.faa.gov/regulations\\_policies/handbooks\\_manuals/aviation/risk\\_management/](https://www.faa.gov/regulations_policies/handbooks_manuals/aviation/risk_management/))

## Deliverable of System code for transfer to AWC:

### Definitions

- A deliverable of system code for transfer to run at the AWC or the Aviation Weather Testbed is defined as an aviation specific function or tool that is purposed for aviation and NWS operations and will be run on local infrastructure at the AWT/AWC in Kansas City, MO.

### Specifications and Documentation

- System code must be built to work Operating Systems used by National Weather Service
  - Common operating systems for the AWC and AWT are Linux RedHat.
  - The current version at AWC is **Linux RedHat version 6**
  - Versions of Linux RedHat do change and different systems with the AWC might run different versions as well. Be sure to describe the version of Linux RedHat you have used to build in your proposal
  - Any libraries used to build your software must be available in the repository for the current version of Linux RedHat being utilized at AWC
- System code must be thoroughly tested before transfer
- Third party libraries may be used but must be requested and approved through the AWC Change Control Board (CCB) which helps to maintain Configuration Management (CM) at the AWC.
  - Third party libraries that are already accepted include NetCDF, HDF5, Python, GNU compilers, bufr, openmotif, and imagemagick.
  - Make sure that any external libraries used are built using the same compiling method as the system code
- It is preferable that the code utilizes a Makefile at the top of the source code directory tree that will build the code from the top down.
  - Makefile should produce a single system executable
- System code will utilize a cron scheduling on AWC systems
- System code should provide descriptive log files in order to fix errors
- Any update patches should be tested against the current operating system
- Any update patches should be tested against any new development versions as well
  - Development versions maybe different from a more recent stable release
- System code must be provided with sufficient documentation for better local support.
  - Support documentation must come in one of the following forms
    - A descriptive README file at the top level of the source code
    - An installation and setup document with build procedures and common bug fixes
    - Documentation must be continually updated
- Make sure the code has been version controlled so that any discrepancies in build procedure may be resolved quickly

## Deliverable of System code for transfer to NCEP via AWC:

### Definitions

- A deliverable of system code for transfer to NCEP via AWC is defined as an aviation specific function or tool that is purposed for aviation and NWS operations and will be run on a national infrastructure outside of the AWC, such as the WCOSS supercomputer in College Park, MD run by NCEP Central Operations (NCO).
- All code delivered for the intent of being run on the national infrastructure will be submitted to the AWC as the “developer” on the national system.

### Specifications and Documentation

- WCOSS runs on a Linux RedHat operating system. Any libraries used must be available in the operating system repository.
  - A third-party library, defined as one not residing in the OS repository, may be requested.
  - Third party libraries that are already accepted include NetCDF, HDF5, Python, GNU compilers, bufr, openmotif, and imagemagick
  - Any third party libraries must be pre-approved. A list of any third-party libraries will be submitted to the AWC.
  - These third-party libraries will be vetted through NCO before use in coding to determine level of support.
  - During the vetting process, these libraries may be installed developmentally
    - When testing against these libraries, make sure they are built using the same compiler with which you build your system code as described in the following section (i.e. intel vs GNU)
- System code must meet the standards for coding as set by NCEP NCO to be run and supported operationally in the specified environment. Such standards include:
  - Easy accessibility of code to NCO staff
  - Sufficient documentation on software building procedures
  - Use of Intel compilers as primary way of building code
    - If using GNU, please specify and be prepared to defend the use
  - Code should be written from an operational perspective as to ease the support process
  - Code should produce descriptive error messages for better support
  - You must have a proposed set of error codes to enable operations staff to be able to more quickly identify an error
  - Code should run on a job submission script. The job submission script may be set to the developers preference (i.e. start the job based on arrival of certain model runs)
- System code must be thoroughly tested before transfer
- Any update patches should be tested against the current operating system
- Any update patches should be tested against any new development versions as well
  - Development versions maybe different from a more recent stable release

## Procedure

- The AWC will work in coordination with the submitting development team to ensure the proper build procedures are complete and work properly in a developmental mode
- Once code is production ready, the AWC will complete the Request for Change (RFC) notification in order to begin transition to production.
  - Once in production mode, any update or patch must be tested developmentally and will require a new RFC form to be implemented
- More detailed coding standards for developers can be obtained by request from either AWC or NCO. Other information and documentation regarding coding libraries such as those allowed in WCOSS and general WCOSS information can be obtained online (<http://www.emc.ncep.noaa.gov/mtt/wcoss.php>)