

InflowWind User's Guide

Andy Platt, Bonnie Jonkman, and Jason Jonkman

July 26, 2016

National Wind Technology Center

Introduction

[InflowWind](#) is a module for processing wind-inflow that has been coupled into the [FAST version 8](#) multi-physics engineering tool to enable aero-elastic simulation of horizontal-axis wind turbines. InflowWind can also be driven as a standalone code to process wind-inflow uncoupled from FAST. This documentation pertains to the latest release of InflowWind—version 3, which represents a major overhaul from earlier version of InflowWind. InflowWind v3 follows the requirements of the [FAST modularization framework](#).

InflowWind supports several wind file formats including uniform (formerly called “hub-height”), binary TurbSim full-field (FF), binary Bladed-style FF, and HAWC formatted binary FF wind files. Additionally, InflowWind supports an internally calculated steady wind model. Unlike previous versions, InflowWind v3 supports arbitrary wind directions for all wind file types.

At each time step, InflowWind receives from the driver code (either the standalone driver or the FAST glue code) the coordinate position of various points and InflowWind returns the undisturbed wind-inflow velocities at these positions. There are no states in the module: each wind velocity component is calculated as a function of the input coordinate positions and internal time-varying parameters, undisturbed from interaction with the wind turbine. For FF wind data types, InflowWind uses Taylor’s frozen turbulence hypothesis—valid only for stationary conditions—to translate wind defined in two-dimensional planes into three spatial dimensions, using the mean wind speed as the advection speed.

This guide describes the various input files of both the InflowWind module and the standalone driver code, as well as the outputs and the formats of the various relevant wind file types. All module inputs and outputs use the SI system (m, s) and are expressed in the global coordinate system illustrated in Figure 1. Angles are assumed to be in radians unless otherwise specified.

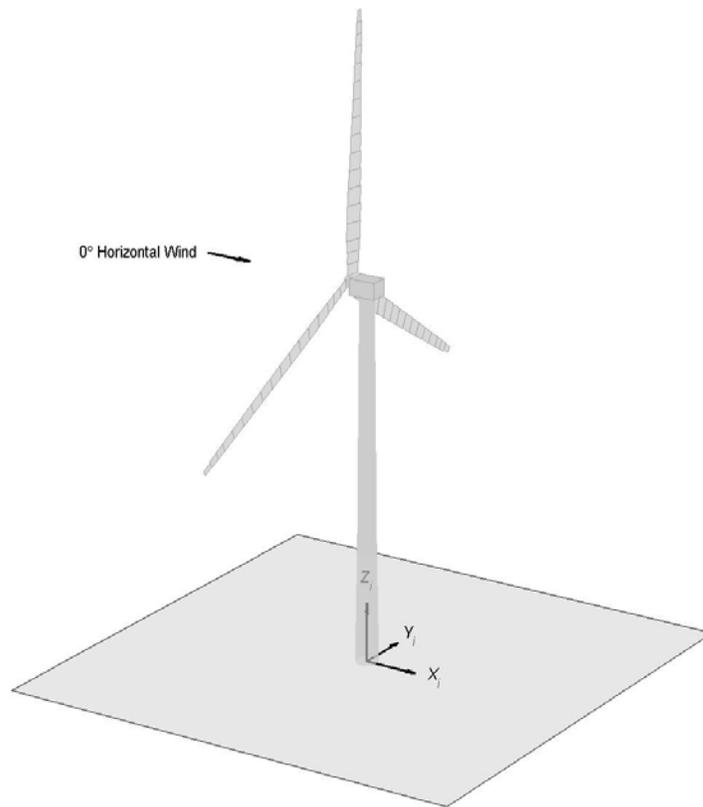


Figure 1: The global coordinate system used in InflowWind

Module Input Files

Primary InflowWind Input file

The parameters in InflowWind's primary input file are divided into several sections (discussed below). The text of an example InflowWind input file is included in Appendix A. Please note that there are multiple inputs with the same name in different sections of this input file (for example, **RefHt**). The parameters in each section are specific to the section in which they are defined. For example, the **RefHt** used in Steady Wind Conditions section is used only for the steady wind calculations; the **RefHt** in the HAWC-format section is used only for HAWC-style files.

General Parameters

The InflowWind primary input file begins with two lines of header information that are yours to use, but are not used by the software. Following the header section is a separator line delineating the main input section that applies to all wind types.

Echo: Echo input data to <RootName>.ech [T/F]

This True/False input parameter is useful for debugging purposes. Set to True if you wish to have the input file contents written to an echo file when the file is read.

WindType: Type of wind file to use [1,2,3,4,5,6]

This switch dictates which type of wind will be used. Options include:

1. Steady wind (calculated internally using the Steady Wind Conditions section)
2. Uniform wind (read externally from a Uniform Wind File)
3. Binary TurbSim FF wind file
4. Binary Bladed-style FF wind file
5. Binary HAWC-style FF wind files
6. User defined using the UserWind subroutine (to be supplied by and compiled by the end user; template *IfW_UserWind.f90* file is provided.)

During linearization analyses with InflowWind coupled with FAST, **WindType** must be 1 (Steady) or 2 (Uniform).

PropagationDir: Direction of wind propagation [degrees]

This input parameter is used to rotate the entire wind field for all wind file types. The rotation angle is specified as a meteorological rotation angle about the Z_i -axis (rotation from alignment with positive X_i towards negative Y_i). This is a clockwise rotation when looking from above towards the surface of the ground or ocean. The local coordinate system, where the wind data is processed, is related to the global coordinate system by (after InflowWind processes the wind data in the local coordinate system, InflowWind will convert the data to the global coordinate system for output):

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos(\text{PropagationDir}) & -\sin(\text{PropagationDir}) & 0 \\ \sin(\text{PropagationDir}) & \cos(\text{PropagationDir}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix}$$

NWindVel: Number of points to output the wind velocity at [-]

Independent of input coordinate positions input to InflowWind from the driver code, you can choose to output the wind velocity at each simulation time step for up to 9 fixed points. **NWindVel** determines the number of points (0-9) to output.

WindVxiList: List of coordinates in the inertial X-direction [m]

This is a white space or comma separated list of the X_i coordinates for each of the **NWindVel** points. If **NWindVel** = 0, this value is ignored.

WindVyiList: List of coordinates in the inertial Y-direction [m]

This is a white space or comma separated list of the Y_i coordinates for each of the **NWindVel** points. If **NWindVel** = 0, this value is ignored.

WindVziList: List of coordinates in the inertial Z-direction [m]

This is a white space or comma separated list of the Z_i coordinates for each of the **NWindVel** points. If **NWindVel** = 0, this value is ignored.

Parameters for steady wind conditions [WindType = 1]

The next section is used only when **WindType** = 1. When steady winds are selected, the parameters in this section are used to calculate a constant wind conditions using a power-

law wind-shear profile. It uses the UniformWind sub-module of InflowWind to calculate the profile. The equation is:

$$V_h(z) = HWindSpeed \left(\frac{z}{RefHt} \right)^{PLexp}$$

HWindSpeed: Horizontal wind speed [m/s]

This is the horizontal wind speed at **RefHt**.

RefHt: Reference height for the horizontal wind speed [m]

This is the reference height for the wind file.

PLexp: Power-law exponent [-]

This is the power-law exponent for the wind shear profile.

Parameters for uniform wind files [*WindType = 2*]

This section is used for reading in uniform wind files (formerly called “hub-height” wind files). See the “Uniform Wind Files” section of this document for information on this type of file.

Filename: Name of the time series data file [-]

This is a quoted string with the name of the uniform wind file to use. The filename can include an absolute or relative path.

RefHt: Reference height for the horizontal wind speed [m]

This is the reference height used in the uniform wind file.

RefLength: Reference length for linear horizontal and vertical shear [m]

This is the reference length used for scaling shear events in the uniform wind file. In previous versions of InflowWind, **RefLength** was set to the rotor diameter.

Parameters for binary TurbSim full-field files [*WindType = 3*]

This section is used for specifying FF wind files generated by TurbSim. See the Binary TurbSim Full-Field Files section of this document for further details about these files.

Filename: name of the full-field TurbSim file to read in [-]

This is the name, in quotes, of the FF binary TurbSim file to read in. Typically this file has a *.bts* file extension. The filename can include an absolute or relative path.

Parameters for binary Bladed-style full-field files [*WindType = 4*]

This section is used for specifying FF wind files generated by TurbSim or Bladed using the Bladed FF wind file format. Two files should exist as a pair; a summary file (*.sum*) and the binary wind file (*.wnd*). Optionally, a tower file may also be used (*.twr*). See the Binary Bladed-Style Full-Field Files section of this document for further details about these files.

FilenameRoot: root name of the full-field Bladed-style file to read in [-]

This is the root filename, in quotes, of the FF binary Bladed-style file to read in without file extension. InflowWind will read **<FilenameRoot>.sum** and **<FilenameRoot>.wnd**. The filename can include an absolute or relative path.

TowerFile: flag indicating the presence of a tower file [T/F]

The tower file includes wind information along the tower. This is an optional file; if **TowerFile** is True, InflowWind will read **<FilenameRoot>.twr** (generated by TurbSim).

Parameters for binary HAWC full-field files [WindType = 5]

This section applies to HAWC FF wind files. HAWC-style binary files come in sets of three: one file for each wind component. The “Binary HAWC-Style Full-Field Files” section of this document contains more details about these files.

FileName_u: name of the file containing the u-component fluctuating wind [-]

This is the name of the file, in quotes, that contains the u-component wind speeds in HAWC-style format, including the *.bin* extension. The filename can include an absolute or relative path.

FileName_v: name of the file containing the v-component fluctuating wind [-]

This is the name of the file, in quotes, that contains the v-component wind speeds in HAWC-style format, including the *.bin* extension. The filename can include an absolute or relative path.

FileName_w: name of the file containing the w-component fluctuating wind [-]

This is the name of the file, in quotes, that contains the w-component wind speeds in HAWC-style format, including the *.bin* extension. The filename can include an absolute or relative path.

nx: number of grids in the x-direction [-]

This is the number of grid points in the x-direction for the three data files listed above.

ny: number of grids in the y-direction [-]

This is the number of grid points in the y-direction for the three data files listed above.

nz: number of grids in the z-direction [-]

This is the number of grid points in the z-direction for the three data files listed above.

dx: distance between points in the x-direction [m]

This is the distance between two consecutive data points in the x-direction in each of the three data files.

dy: distance between points in the y-direction [m]

This is the distance between two consecutive data points in the y-direction in each of the three data files.

dz: distance between points in the z-direction [m]

This is the distance between two consecutive data points in the z-direction in each of the three data files.

RefHt: Reference height for HAWC wind files [m]

The reference height specifies the vertical center of the wind grid.

Scaling parameters for turbulence

This is a subsection of the HAWC-format binary files section. HAWC-style data files are typically scaled to achieve a particular standard deviation. The parameters in this section determine how the raw data in the three HAWC-style data files will be scaled.

ScaleMethod: turbulence scaling method [0,1,2]

There are several methods for scaling the turbulence including the following:

0. none – no scaling applied
1. direct scaling
2. calculate scaling based on desired standard deviation

The direct scaling option (***ScaleMethod*** = 1) scales all of the points in the file by a particular scaling factor, specified by ***SF_x***, ***SF_y***, and ***SF_z***. When ***ScaleMethod*** = 2, InflowWind calculates the scaling factors necessary to achieve the requested standard deviations, ***SigmaF_x***, ***SigmaF_y***, and ***SigmaF_z***. InflowWind calculates the standard deviation of each wind component at one point in roughly the center of the grid (at points $FLOOR((nz+1)/2)$ and $FLOOR((ny+1)/2)$), using all of the points in the x direction. It then computes the scaling factors necessary to achieve the requested standard deviations. These computed scaling factors are printed in the InflowWind summary file.

SF_x: Turbulence scaling factor for x-direction [-] [ScaleMethod=1]

This is the scaling factor for u component if direct scaling is used. All of the wind-speed values in ***FileName_u*** will be multiplied by ***SF_x***.

SF_y: Turbulence scaling factor for y-direction [-] [ScaleMethod=1]

This is the scaling factor for v component if direct scaling is used. All of the wind-speed values in ***FileName_v*** will be multiplied by ***SF_y***.

SF_z: Turbulence scaling factor for z-direction [-] [ScaleMethod=1]

This is the scaling factor for w component if direct scaling is used. All of the wind-speed values in ***FileName_w*** will be multiplied by ***SF_z***.

SigmaF_x: Turbulence standard deviation for x-direction [m/s] [ScaleMethod=2]

This is the requested standard deviation for the u-component wind speed if standard deviation scaling is used.

SigmaF_y: Turbulence standard deviation for y-direction [m/s] [ScaleMethod=2]

This is the requested standard deviation for the v-component wind speed if standard deviation scaling is used.

SigmaFz: Turbulence standard deviation for z-direction [m/s] [ScaleMethod=2]

This is the requested standard deviation for the w-component wind speed if standard deviation scaling is used.

Mean wind profile parameters

This is a subsection of the HAWC-format binary files section. Because the HAWC-style wind files do not contain a mean wind speed, InflowWind superimposes a mean wind-speed profile, added to the u-component wind speed.

URef: Mean u-component wind speed at the reference height [m/s]

This is the mean wind speed at **RefHt**.

WindProfile: Wind profile type [0,1,2]

There are three possible wind-profile types available:

0. Constant
1. Logarithmic
2. Power law

For the constant wind profile,

$$u(z) = URef$$

The logarithmic wind profile is

$$u(z) = URef \frac{\ln \frac{z}{Z0}}{\ln \frac{RefHt}{Z0}}$$

The power-law wind profiles is

$$u(z) = URef \left(\frac{z}{RefHt} \right)^{PLexp}$$

PLexp: power law exponent [-]

This is the power-law exponent, used only when **WindProfile** = 2.

Z0: surface roughness length [m]

The surface roughness length—a measure of the roughness of the surface terrain—is the extrapolated height at which the mean wind speed becomes zero in a neutral atmosphere, assuming a logarithmic vertical wind profile. It is used only when **WindProfile** = 1.

Output section

SumPrint: output a summary file [T/F]

When set to True, a summary file with information regarding the selected wind type is written. This file will have the filename of <RootName>.IfW.sum where *RootName* is the root name of the InflowWind driver input file (when run in standalone mode) or the root name of the FAST input file (when coupled with FAST).

Outlist: list of output parameters to add to the FAST output file [-]

When InflowWind is used in FAST, the **OutList** section controls output quantities generated by InflowWind to the FAST output files. Enter one or more lines containing quoted strings that in

turn contain one or more output parameter names. Separate output parameter names by any combination of commas, semicolons, spaces, and/or tabs. If you prefix a parameter name with a minus sign, “-”, underscore, “_”, or the characters “m” or “M”, InflowWind will multiply the value for that channel by -1 before writing the data. The parameters are written in the order they are listed in the input file. InflowWind allows you to use multiple lines so that you can break your list into meaningful groups and so the lines can be shorter. You may enter comments after the closing quote on any of the lines. Entering a line with the string “END” at the beginning of the line or at the beginning of a quoted string found at the beginning of the line will cause InflowWind to quit scanning for more lines of channel names. If InflowWind encounters an unknown/invalid channel name, it warns the users but will remove the suspect channel from the output file. Please refer to Appendix B for a complete list of possible output parameters.

Uniform Wind Files

Uniform wind files (formally called “hub-height” wind files) are formatted (human-readable text) files made up of eight columns of data that are used to create uniform, but time-varying, wind conditions. These columns are described in Table 1.

Table 1: Data Columns in Uniform Wind Files

Column	Description	Units
1 Tdat	Time	s
2 Vh_ref	Horizontal wind speed at RefHt vectorial $u+v$	m/s
3 Delta	Wind direction, positive clockwise looking down	deg
4 VZ	Vertical wind speed (w component)	m/s
5 HLinShr	Horizontal linear wind-shear parameter	-
6 VShr	Vertical power-law wind-shear exponent	-
7 VLinShr	Vertical linear wind-shear parameter	-
8 VGust	Horizontal gust speed	m/s

Uniform wind files may contain any number of comment lines at the beginning of the file. Each comment line must contain at least one of the characters “!”, “#”, or “%” (excluding the quotes). These characters are normally at the beginning of the comment lines, but they need not be. The first line *not* containing any of the comment characters is considered the start of the wind data lines in the file. No comment lines are allowed after the first data line.

The wind data columns in the file must be separated by tabs, spaces, or commas. There is one line for each time, but the time step between lines need not be constant. When the simulation time falls between two **Tdat** values in the uniform wind file, the *columns* (not the wind speeds computed from them) are linearly interpolated to get wind data values for that simulation time. If the simulation runs longer than the last time in the uniform wind file, the simulation continues using the last line of the file for the remainder of the simulation.

The horizontal and vertical wind speeds are computed using the (possibly interpolated) wind data as follows:

$$\begin{aligned}
 V_h(x, y, z) &= V_{h_{ref}} \left(\frac{z}{RefHt} \right)^{VShr} \\
 &+ V_{h_{ref}} \left(\frac{HLinShr}{RefLength} \right) (x \sin(Delta) + y \cos(Delta)) \\
 &+ V_{h_{ref}} \left(\frac{VLinShr}{RefLength} \right) (z - RefHt) + V_{Gust}
 \end{aligned}$$

$$V_z(x, y, z) = VZ$$

where **RefLength** and **RefHt** are values set in the InflowWind input file (see “Parameters for uniform wind files [$WindType = 2$]”)

Binary TurbSim Full-Field Files

TurbSim binary FF files are 3-dimensional data sets (functions of \mathbf{y} , \mathbf{z} , and \mathbf{t}) of the 3-component wind inflow velocity, \mathbf{u} , \mathbf{v} , \mathbf{w} . The time series data are stored in a $NumGrid_Y \times NumGrid_Z$ evenly-spaced grid, $\mathbf{V}(\mathbf{y}, \mathbf{z}, \mathbf{t})$, and—optionally—tower points in a single line below that grid, $\mathbf{V}_{tower}(\mathbf{z}, \mathbf{t})$.

These binary files typically have a “.bts” extension. They contain normalized time-series data encoded as 2-byte integers. The first part of each file is a header that provides information about the grid and tells InflowWind how to convert the integers to floating-point values. The wind speeds for the $NumGrid_Y \times NumGrid_Z$ grids and the tower points (if specified) follow that. The file format is described in detail in Appendix D of this document.

InflowWind uses Taylor’s frozen turbulence hypothesis to translate $\mathbf{V}(\mathbf{y}, \mathbf{z}, \mathbf{t})$ to $\mathbf{V}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{t})$, using the mean wind speed stored in the binary file as the advection speed. The grids are interpolated using a tri-linear interpolation scheme. The tower points are interpolated using a bi-linear interpolation scheme, assuming that \mathbf{V}_{tower} does not vary with \mathbf{y} . InflowWind will throw a fatal error if the coordinate where it is requested to calculate a wind speed is outside the domain of the binary wind file (i.e., it does not extrapolate).

When the binary file indicates that the data it contains is periodic (i.e., it contains a full period), InflowWind starts the first data point at $\mathbf{x}=\mathbf{0}$, and the wind file is treated as periodic in time so it never exceeds the bounds in the x-direction or time.

When the binary file is not periodic, data in the file are shifted so that the first data point is half the grid width ahead of the turbine, $\mathbf{x}=-\mathbf{(ny-1)*dy}/2$.

Binary Bladed-Style Full-Field Files

InflowWind can read Bladed-style binary FF files in the format generated by TurbSim. Wind files generated by Bladed can be read, but it should be noted that InflowWind does not add a mean wind profile.

The binary wind data file (.wnd) must always have a text summary file (.sum). Optionally, a tower file corresponding to the wind speeds on a line beneath the grid in the wind data file may also be used (.twr). These three file types are described below.

InflowWind uses the same interpolation method for the Bladed-style full-field files as the TurbSim full-field files: Taylor’s frozen turbulence hypothesis is used to translate $\mathbf{V}(\mathbf{y}, \mathbf{z}, \mathbf{t})$ to $\mathbf{V}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{t})$, using the mean wind speed stored in the binary file as the advection speed. The grids are interpolated using a tri-linear interpolation scheme. The tower points are interpolated using a bi-linear interpolation scheme, assuming that \mathbf{V}_{tower} does not vary with \mathbf{y} . InflowWind will throw a fatal error if the coordinate where it is requested to calculate a wind speed is outside the domain of the binary wind file (i.e., it does not extrapolate).

When the summary file indicates that the data it contains is periodic (i.e., it contains a full period), InflowWind starts the first data point at $x=0$, and the wind file treated as periodic in time so it never exceeds the bounds in the x-direction or time.

When the summary file does not contain the periodic key word, data in the file are shifted so that the first data point is half the grid width ahead of the turbine, $x=-(ny-1)*dy/2$.

Wind Data File

The wind data files are binary files that must have a “.wnd” extension. They contain normalized time-series data encoded as 2-byte integers. The first part of each file is a header that provides some information about the grid. The wind speeds for the $NumGrid_Y \times NumGrid_Z$ grids follow that. The description of this file format is contained in Appendix E.

Summary File

To read the binary Bladed-style FF files, InflowWind needs to know some summary information about the grids and how they were scaled. If the Bladed-style binary files were generated by TurbSim, this information is contained in the TurbSim-generated summary file. If, however, the files were generated by another program, you will need to generate a summary file. The description of this file format is contained in Appendix F.

Tower Wind File

TurbSim has the ability to generate winds along a central vertical line below the grid, if requested. This file has the same root name as the wind data file and summary file, and it contains a “.twr” extension. This binary file contains a header and 2-byte-integer data that are normalized in the same way as the wind data on the grid. The description of this file format is contained in Appendix G.

Binary HAWC-Style Full-Field Files

HAWC FF files are 3-dimensional data sets (functions of x , y , and z) of the 3-component wind inflow velocity, u , v , w . The data are stored in a $nx \times ny \times nz$ evenly-spaced grid, $V(x,y,z)$.

HAWC-style binary files do not contain any header information. All data necessary to read and scale it must be entered in the InflowWind input file. Each data file contains the wind speed for a specific wind component, stored as 4-byte real numbers.

```
ix = nx, nx-1, ... 1
  iy = ny, ny-1, ... 1
    iz = 1, 2, ... nz
      Vgrid(iz,iy,ix,i)
    end iz
  end iy
end it
```

After the grids are read, the turbulence for each component is scaled according to the method selected in the InflowWind input file. After this scaling, the mean wind profile is added to the u-component wind speed.

InflowWind uses Taylor's frozen turbulence hypothesis to translate the HAWC turbulence box, $V(x,y,z)$ in time using the mean wind speed, U_{Ref} , from the InflowWind input file as the advection speed. The grids are interpolated using a tri-linear interpolation scheme. InflowWind will throw a fatal error if the coordinate where it is requested to calculate a wind speed is outside the domain of the binary wind file (i.e., it does not extrapolate).

InflowWind starts the first data point at $x=0$, and the wind file treated as periodic in time so it never exceeds the bounds in the x-direction or time.

Standalone Driver for InflowWind

The standalone InflowWind program, *InflowWind_Driver_Win32.exe*, processes wind files without coupling to FAST. The standalone version requires the use of a driver input file in addition to the primary InflowWind input file. This driver file specifies initialization inputs normally provided to InflowWind by FAST, as well as the time step inputs to InflowWind.

The standalone driver code was developed primarily as a tool for testing InflowWind during development, and as such its capabilities are somewhat limited, but it may be useful tool for analyzing wind data that will be used in a FAST simulation. Wind data can be output at a set of user-specified points or in a gridded volume.

Run the standalone InflowWind software from a DOS prompt by typing, e.g.

```
> InflowWind_Driver_Win32.exe <DriverIptFile.dvr>
```

Driver Input File

The parameters in InflowWind's driver input file are divided into several sections (discussed below). The text of an example InflowWind driver input file is included in Appendix C of this guide. An overview of the different parameters in the driver input file follows.

Echo: Echo input data to <RootName>.ech [T/F]

This True/False input parameter is useful for debugging purposes. Set to True if you wish to have the input file contents written to an echo file when the file is read.

IfWFileName: Name of the InflowWind input file [-]

This is the name of the input file for InflowWind.

NumTSteps: Number of time steps to run [-]

This is the number of time steps to take when making calls to InflowWind. The word 'default' may be used to call InflowWind at each time step within the selected wind file. The

number of timesteps may be adjusted within the driver so as to not overrun the end of the file when 'default' is chosen.

TStart: start time of the simulation [s]

This is the time to call InflowWind for the first timestep.

Summary: summarize the data extents in the wind file [T/F]

This True/False option is currently disabled.

SummaryFile: write a summary to file (<rootname>.dvr.sum) [T/F]

This True/False option is currently disabled.

DT: timestep [s]

This sets the time step InflowWind will use. The word 'default' may be used to call InflowWind with the time step within the selected wind file (note that this option is not allowed for uniform wind files with unequally spaced timesteps).

PointsFile: read a points file and return wind velocity at each timestep [T/F]

This True/False input parameter indicates if a points file should be processed.

PointsFileName: name of the points file [-]

This is the name of the points file to read (quoted string). The points file contains a series of user-specified X_i, Y_i, Z_i coordinates; InflowWind will calculate the wind velocity at each time step for each point. Results are written to $\langle PointsFileName \rangle.Velocity.dat$. The points file may contain an arbitrary number of header lines starting with the characters #, %, or !. Coordinates must appear in three white space or comma delimited columns.

WindGrid: extract wind data for a volume of X,Y,Z coordinates [T/F]

This True/False flag indicates that the wind velocities for a gridded volume of coordinates will be written to an output file ($\langle RootName \rangle.WindGrid.out$). The exported data will be arranged in blocks of time corresponding to the range of time set by ***TStart***, ***NumTSteps***, and ***DT***. The volume is defined by the following sets of parameters.

GridCtrCoord: the coordinate of the center of the volume [m]

The X_i, Y_i, Z_i coordinates for the center of the volume.

GridDX, GridDY, GridDZ: stepsize of the grid [m]

The spacing between the points in the $X_i, Y_i,$ and Z_i dimensions. A value of zero for a dimension will flatten that dimension to a single point.

GridNX, GridNY, GridNZ: number of grid points [-]

The number of grid points in the $X_i, Y_i,$ and Z_i dimensions. A value of zero for a dimension will flatten that dimension to a single point.

Command line options

Several command line options are built into the standalone InflowWind driver. A short description of them is available by typing

```
> InflowWind_Driver_Win32.exe /help
```

at the DOS command prompt. The help menu is as follows:

```
Syntax: InflowWind_Driver <filename> [options]
```

```
where:    <filename>    -- Name of driver input file to use
options:  /ifw          -- treat <filename> as name of InflowWind input file
                    (no driver input file)
```

```
The following options will override values in the driver input file:
```

```
/DT[#]    -- timestep
/TStart[#] -- start time
/TSteps[#] -- number of timesteps
/xrange[#:#] -- range of x (#'s are reals)
/yrange[#:#] -- range of y
/zrange[#:#] -- range in z (ground = 0.0)
/Dx[#]    -- spacing in x
/Dy[#]    -- spacing in y
/Dz[#]    -- spacing in z
/points[FILE] -- calculates at x,y,z coordinates specified in a
                    white space delimited FILE
/v        -- verbose output
/vv       -- very verbose output
/help     -- print this help menu and exit
```

```
Notes:
```

```
-- Unspecified ranges and resolutions default to what is in the file.
-- If no XRange is specified, assumed to be only at X=0
-- Options are not case sensitive.
```

Any command line options will override values found in the driver input file. Note that the syntax for the options on Windows operating system use the '/' character, whereas the command line options the InflowWind driver program on a Mac or Linux operating system use the '-' character.

/ifw: no driver input file

The use of this option will cause the driver code to treat the filename provided as the InflowWind input file name.

/dt[#]: timestep [s]

This sets the time step InflowWind will use.

/tstart[#]: start time [s]

This sets the time of the first time step of the file reading.

/xrange[#:#]: range in x to extract gridded data from [m]

This sets the X_i -axis range over which gridded wind data will be extracted and written to a <filename>.WindGrid.out file.

/yrange[#:#]: range in y to extract gridded data from [m]

This sets the Y_i -axis range over which gridded wind data will be extracted and written to a <filename>.WindGrid.out file.

/zrange[#:#]: range in z to extract gridded data from [m]

This sets the Z_i -axis range over which gridded wind data will be extracted and written to a *<filename>.WindGrid.out* file.

/dx[#]: points spacing along x direction

This sets the X_i -axis point spacing for the gridded wind data will be extracted and written to a *<filename>.WindGrid.out* file.

/dy[#]: points spacing along y direction

This sets the Y_i -axis point spacing for the gridded wind data will be extracted and written to a *<filename>.WindGrid.out* file.

/dz[#]: points spacing along z direction

This sets the Z_i -axis point spacing for the gridded wind data will be extracted and written to a *<filename>.WindGrid.out* file.

/points[FILE]: read a points FILE and return wind velocity at each timestep

Read a file containing a series of user-specified X_i, Y_i, Z_i coordinates and calculate the wind velocity at each time step for each point. Results are written to *<FILE>.Velocity.dat* file. The points file may contain an arbitrary number of header lines starting with the characters #, %, or !. Coordinates must appear in three white space or comma delimited columns.

/v: verbose

This option returns extra information on the internal settings of the standalone InflowWind program.

/vv: very verbose

This option returns even more extra information on the internal settings of the standalone InflowWind program, error status after calls to InflowWind, information on the coordinate transformations for rotating the wind field, and mapping of ***OutList*** parameters from the InflowWind primary input file.

Appendix A: InflowWind Primary Input File Format – Steady Wind Example

```

----- InflowWind v3.00.* INPUT FILE -----
This is a sample Inflow-Wind input file.
-----
True      Echo      - Echo input data to <RootName>.ech (flag)
1         WindType  - switch for wind file type (switch) [1=steady; 2=uniform; 3=binary TurbSim FF; 4=binary Bladed-style FF; 5=HAWC format; 6=User...]
60.0     PropogationDir - Direction of wind propogation (meteorological rotation from aligned with X (positive rotates towards -Y) -- degrees)
2        NWindVel  - Number of points to output the wind velocity (0 to 9)
0,10     WindVxIList - List of coordinates in the inertial X direction (m)
0,0      WindVyIList - List of coordinates in the inertial Y direction (m)
50,50    WindVzIList - List of coordinates in the inertial Z direction (m)
===== Parameters for Steady Wind Conditions [used only for WindType = 1] =====
10.0     HWindSpeed - Horizontal windspeed (m/s)
90.0     RefHt    - Reference height for horizontal wind speed (m)
0.14     PLExp   - Power law exponent (-)
===== Parameters for Uniform wind file [used only for WindType = 2] =====
"Test001.hh"  Filename - Filename of time series data for uniform wind field (-)
90.0         RefHt    - Reference height for horizontal wind speed (m)
100.0       RefLength - Reference length for linear horizontal and vertical sheer (m)
===== Parameters for Binary TurbSim Full-Field files [used only for WindType = 3] =====
"          "  Filename - Name of the Full field wind file to use (.bts)
===== Parameters for Binary Bladed-style Full-Field files [used only for WindType = 4] =====
"          "  FilenameRoot - Name of the Full field wind file to use (.wnd, .sum)
T          TowerFile  - Have tower file (.twr) (flag)
===== Parameters for HAWC-format binary files [Only used with WindType = 5] =====
"basic_5u.bin"  FileName_u - name of the file containing the u-component fluctuating wind (.bin)
"basic_5v.bin"  FileName_v - name of the file containing the v-component fluctuating wind (.bin)
"basic_5w.bin"  FileName_w - name of the file containing the w-component fluctuating wind (.bin)
64            nx      - number of grids in the x direction (in the 3 files above) (-)
32            ny      - number of grids in the y direction (in the 3 files above) (-)
32            nz      - number of grids in the z direction (in the 3 files above) (-)
16            dx      - distance between points in the x direction (m)
3             dy      - distance between points in the y direction (m)
3             dz      - distance between points in the z direction (m)
50            RefHt   - reference height; the height of the vertical center of the grid (m)
----- Scaling parameters for turbulence -----
1           ScaleMethod - Turbulence scaling method (switch) [0=none, 1=direct scaling, 2=calculate scaling factor based on a desired standard deviation]
1.0        SFx        - Turbulence scaling factor for the x direction (-) [ScaleMethod=1]
1.0        SFy        - Turbulence scaling factor for the y direction (-) [ScaleMethod=1]
1.0        SFz        - Turbulence scaling factor for the z direction (-) [ScaleMethod=1]
12.0       SigmaFx    - Turbulence standard deviation to calculate scaling from in x direction (m/s) [ScaleMethod=2]
8.0        SigmaFy    - Turbulence standard deviation to calculate scaling from in y direction (m/s) [ScaleMethod=2]
2.0        SigmaFz    - Turbulence standard deviation to calculate scaling from in z direction (m/s) [ScaleMethod=2]
----- Mean wind profile parameters (added to HAWC-format files) -----
5.0        URef       - Mean u-component wind speed at the reference height (m/s)
2          WindProfile - Wind profile type (switch) [0=constant; 1=logarithmic, 2=power law]
0          PLExp      - Power law exponent (-) (used for PL wind profile type only)
0.03      Z0          - Surface roughness length (m) (used for LG wind profile type only)
===== OUTPUT =====
True      SumPrint   - Print summary data to <RootName>.sum (flag) (currently unused)
          OutList    - The next line(s) contains a list of output parameters. See OutListParameters.xlsx for a listing of available output channels, (-)
"Wind1VelX,Wind1VelY,Wind1VelZ" - Wind velocity at point WindVxIList(1),WindVyIList(1),WindVzIList(1). X, Y, and Z direction components.
"Wind2VelX,Wind2VelY,Wind2VelZ" - Wind velocity at point WindVxIList(2),WindVyIList(2),WindVzIList(2). X, Y, and Z direction components.
END of input file (the word "END" must appear in the first 3 columns of this last OutList line)

```

Appendix B: List of Output Channels

This is a list of all possible output parameters for the InflowWind module.

Channel Name	Units	Description
Wind1VelX, Wind2VelX, Wind3VelX, Wind4VelX, Wind5VelX, Wind6VelX, Wind7VelX, Wind8VelX, Wind9VelX	(m/s)	Wind velocity in the horizontal X _i -axis (at up to 9 fixed points designated by <i>NWindVel</i> , <i>WindVxiList</i> , <i>WindVyiList</i> , and <i>WindVziList</i>)
Wind1VelY, Wind2VelY, Wind3VelY, Wind4VelY, Wind5VelY, Wind6VelY, Wind7VelY, Wind8VelY, Wind9VelY	(m/s)	Wind velocity in the horizontal Y _i -axis (at up to 9 fixed points designated by <i>NWindVel</i> , <i>WindVxiList</i> , <i>WindVyiList</i> , and <i>WindVziList</i>)
Wind1VelZ, Wind2VelZ, Wind3VelZ, Wind4VelZ, Wind5VelZ, Wind6VelZ, Wind7VelZ, Wind8VelZ, Wind9VelZ	(m/s)	Wind velocity in the vertical Z _i -axis (at up to 9 fixed points designated by <i>NWindVel</i> , <i>WindVxiList</i> , <i>WindVyiList</i> , and <i>WindVziList</i>)

Appendix C: Example InflowWind Driver Input File Format

```
Test number 004
InflowWind driver input file. V1.00
  t          echo (flag)
-----
"Test004--BladedFF.ipt"  IfWFileName  -- IfW input filename (-)
  DEFAULT  NumTSteps    -- number of timesteps to run (DEFAULT for all)      (-)
  0.0      TStart       -- Start time                               (s)
  f        Summary     -- Summarize the data extents in the windfile (flag)
  f        SummaryFile -- Write summary to file .dvr.sum (flag)
---- IfW inputs -----
  0.0025   DT          -- timestep size for driver to take (s, or DEFAULT for what the file contains)
---- Points file input (output given as PointsFileName.Velocity.dat) -----
  t        PointsFileName -- read in a list of points from a file (flag)
"Test004--Points.txt"  PointsFileName -- name of points file (-) (comma separated x,y,z coordinates, # symbol for comments)
---- Output grid (Points below ground will simply be ignored) -----
  T        WindGrid    -- report wind data at set of Y,Z coordinates (flag)
  6,0,15  GridCtrCoord -- coordinates of center of grid                (m)
  1,1,0   GridDx,GridDY,GridDZ -- Stepsize of grid                (m)
  1,1,0   GridNx,GridNY,GridNZ -- number of grid points in X, Y and Z directions (-)
END of driver input file
```

Appendix D: Format for Binary TurbSim Full-Field Files

These binary files begin with a header, which is described in Table 2.

Table 2: TurbSim binary file header information

Type (Bytes)	Parameter	Description
Integer (2)	ID	Identifies the file as a TurbSim binary file. ID should have the value 7 (not periodic) or 8 (periodic).
Integer (4)	$NumGrid_Z$	The number of grid points in the vertical direction
Integer (4)	$NumGrid_Y$	The number of grid points in the horizontal direction
Integer (4)	n_{tower}	The number of tower points below the grid
Integer (4)	n_t	The number of time steps
Real (4)	dz	The distance in meters between two adjacent points in the vertical direction
Real (4)	dy	The distance in meters between two adjacent points in the horizontal direction
Real (4)	$TimeStep$	The time in seconds between consecutive grids
Real (4)	u_{hub}	The mean wind speed in m/s at hub height
Real (4)	$HubHt$	The height in meters of the hub (or reference point)
Real (4)	Z_{bottom}	The height in meters of the bottom of the grid
<i>for i = 1, 2, 3</i>		
Real (4)	$V_{slope}(i)$	The slope used to scale the i^{th} velocity component ¹ from 4-byte reals into 2-byte integers.
Real (4)	$V_{intercept}(i)$	The intercept used to scale the i^{th} velocity component ¹ from 4-byte reals into 2-byte integers.
<i>end i</i>		
Integer (4)	$n_{characters}$	The number of characters in the ASCII string that gives the TurbSim version number, date, and time the file was generated.
<i>for i = 1, 2, ... n_characters</i>		
Integer (1)	$Character_i$	The ASCII integer representation of the i^{th} character of the string that gives the TurbSim version number, date, and time the file was generated. ACHAR($Character_i$) gives the character.
<i>end i</i>		

¹ The three wind components are defined as $u = 1$, $v = 2$, and $w = 3$.

Following the header are the normalized data for the grids and tower, stored as 2-byte integers. The wind speed component (u,v,w) is looped fastest, followed by grid point location, and time is looped over slowest:

```

it = 1, 2, ... nt
  iz = 1, 2, ... NumGrid_Z
    iy = 1, 2, ... NumGrid_Y
      i = 1, 2, 3
        Vgrid_norm(i,iy,iz,it)
      end i
    end iy
  end iz

  iz = 1, 2, ... ntower
    i = 1, 2, 3
      Vtower_norm(i,iz,it)
    end i
  end iz
end it

```

$V_{grid_norm}(i,iy,iz,it)$ is the normalized i^{th} velocity component of the wind speed at time step, it , and grid location ($Y_{grid}(iy)$, $Z_{grid}(iz)$), and $V_{tower_norm}(i,iz,it)$ is the normalized i^{th} -component of the wind speed at time step, it , and tower height, $Z_{tower}(iz)$.

To convert the normalized wind in the FF TurbSim binary file to velocities in units of meters per second, use the following equations:

$$V_{grid}(i,iy,iz,it) = \frac{V_{grid_norm}(i,iy,iz,it) - V_{intercept}(i)}{V_{slope}(i)}$$

$$V_{tower}(i,iz,it) = \frac{V_{tower_norm}(i,iz,it) - V_{intercept}(i)}{V_{slope}(i)}$$

The corresponding lateral locations, Y , and vertical locations, Z , of the grid and/or tower points are given in units of meters by

$$Y_{grid}(iy) = -0.5(NumGrid_Y - 1)dy + (iy - 1)dy$$

$$Z_{grid}(iz) = Z_{bottom} + (iz - 1)dz$$

and

$$Y_{tower} = 0$$

$$Z_{tower}(iz) = Z_{bottom} - (iz - 1)dz$$

Table 4:Format of Grid Velocities in Full-Field Bladed-Style Binary File Format

Type (Bytes)	Parameter	Description
<i>for it = 1, 2, ... nt</i>		
<i> for iz = 1, 2, ... NumGrid_Z</i>		
<i> for iy = 1, 2, ... NumGrid_Y</i>		
Integer (2)	$U_{grid_norm}(iy, iz, it)$	The normalized U component of the wind speed at time step, it , and grid location $(y(iy), z(iz))$.
Integer (2)	$V_{grid_norm}(iy, iz, it)$	The normalized V component of the wind speed at time step, it , and grid location $(y(iy), z(iz))$.
Integer (2)	$W_{grid_norm}(iy, iz, it)$	The normalized W component of the wind speed at time step, it , and grid location $(y(iy), z(iz))$.
<i> end iy</i>		
<i>end iz</i>		
<i>end it</i>		

To convert the normalized wind in the FF Bladed-style binary file to velocities in units of meters per second, use the following equations:

$$U_{grid}(iy, iz, it) = u_{hub} \left(\frac{TI(u)}{1000} U_{grid_norm}(iy, iz, it) + 1 \right),$$

$$V_{grid}(iy, iz, it) = u_{hub} \left(\frac{TI(v)}{1000} V_{grid_norm}(iy, iz, it) \right),$$

and

$$W_{grid}(iy, iz, it) = u_{hub} \left(\frac{TI(w)}{1000} W_{grid_norm}(iy, iz, it) \right).$$

Here TI represents the turbulence intensity as a decimal, not a percentage.

The corresponding vertical locations, Z , of the grid points are given in units of meters by

$$Z_{grid}(iz) = -0.5(NumGrid_Z - 1)dz + HubHt - HeightOffset + (iz - 1)dz$$

using values of $HubHt$ and $HeightOffset$ from the summary file. The lateral locations, Y , of the grid points depend on the input value $Clockwise$ (read from the summary file) and are given by

$$Y_{grid}(iy) = \begin{cases} -0.5(NumGrid_Y - 1)dy + (iy - 1)dy, & Clockwise = FALSE \\ 0.5(NumGrid_Y - 1)dy - (iy - 1)dy, & Clockwise = TRUE \end{cases}$$

Appendix F: Format for Summary Files Associated With Bladed-Style Full-Field Files in InflowWind

T/F indicates a LOGICAL input value

NUM indicates a numeric (REAL, floating-point) input value

The file name must end in “.sum” and have the same root name as the binary FF file (which should have a “.wnd” extension).

The words in **bold red font** are required unless otherwise specified, and they must appear in the order listed below.

InflowWind looks for those key words, so the exact spelling and spacing must be used. There may be other lines mixed in the file, with the exception that the three lines following **UBAR** must be the **TI** parameters.

The file should look like the text contained in the box below:

T/F **CLOCKWISE**

NUM **HUB HEIGHT**

NUM **UBAR** (if the character “=” exists, it must precede the numeric entry; e.g., UBar = 12.0 m/s)

NUM TI(u) (if the characters “=” or “%” exists, the numeric entry must follow the “=” and precede the “%”; e.g., TI(u) = 12.42 %)

NUM TI(v) (if the characters “=” or “%” exists, the numeric entry must follow the “=” and precede the “%”; e.g., TI(v) = 9.94 %)

NUM TI(w) (if the characters “=” or “%” exists, the numeric entry must follow the “=” and precede the “%”; e.g., TI(w) = 6.21 %)

NUM **HEIGHT OFFSET** (optional; if the character “=” exists, it must precede the numeric entry)

PERIODIC (optional; if this string exists, the file will be interpolated as if it is periodic in time)

The **Clockwise** flag is used to determine which way the lateral dimension of the grids was written to the binary file, i.e., does it start on the right or the left?

UBar and the **TI** values are used for normalizing the binary data (the **TI** need not be the actual turbulence intensities; **UBar** must be within 0.1 m/s of the mean FF wind speed contained in the binary file). **UBar** must be in units of meters per second (m/s) and **TI** must be entered as a percent.

Height Offset is used to allow grids that are not centered vertically on the turbine hub-height. It must be given in units of meters, and it can be calculated by the equation

$$\text{HeightOffset} = \text{HubHeight} - \text{GridBase} - \text{FFZHWid}.$$

Periodic is used to allow the wind file to be periodic in time when it is interpolated in InflowWind. This option should be used only if the wind file contains a full period. The first grid in periodic files is located at $X = 0 \text{ m}$; the first grid in non-periodic files is located at $X = \frac{\text{GridWidth}}{2} \text{ m} = \frac{(\text{ny}-1)\text{dy}}{2} \text{ m}$.

Appendix G: Format for Tower Files Associated With Bladed-Style Full-Field Files in InflowWind

TurbSim tower files have a “.*twr*” extension. Each file contains a header of 4-byte real and integer values, followed by 2-byte integer time series of the three wind components at each point on the tower grid. The wind components at the tower points are normalized and stored in 2-byte binary integers, exactly the same way that Bladed-style full-field wind files are written. The tower files have the same vertical resolution as the full-field grid, with points going downward from the bottom of the full grid in a single line at the tower centerline.

Table 5: Format of Header in TurbSim Binary Tower-Data File

Type (Bytes)	Parameter	Description
Real (4)	<i>dz</i>	Vertical grid resolution, in meters.
Real (4)	$u_{hub} * TimeStep$	Longitudinal grid resolution, in meters.
Real (4)	<i>Zmax</i>	The height of the highest tower point, in meters.
Real (4)	<i>nt</i>	The number of points in the longitudinal direction.
Real (4)	<i>nz</i>	The number of vertical tower points.
Real (4)	u_{hub}	The mean wind speed, in meters per second.
Real (4)	$100 * TI(u)$	The turbulence intensity of the <i>u</i> component, in percent.
Real (4)	$100 * TI(v)$	The turbulence intensity of the <i>v</i> component, in percent.
Real (4)	$100 * TI(w)$	The turbulence intensity of the <i>w</i> component, in percent.

Table 6: Format of Grid Velocities in TurbSim Binary Tower-Data File

For each increasing time step (*nt* points), and starting at the top of the grid, going downward (*nz* points) the data are stored as:

Type (Bytes)	Parameter	Description
<i>for it = 1, 2, ... nt</i>		
<i>for iz = 1, 2, ... nz</i>		
Integer (2)	$U_{tower_norm}(iz, it)$	Normalized <i>U</i> component of the wind speed at time step, <i>it</i> , and height <i>z(iz)</i> .
Integer (2)	$V_{tower_norm}(iz, it)$	Normalized <i>V</i> component of the wind speed at time step, <i>it</i> , and height <i>z(iz)</i> .
Integer (2)	$W_{tower_norm}(iz, it)$	Normalized <i>W</i> component of the wind speed at time step, <i>it</i> , and height <i>z(iz)</i> .
end iz		

end it

To convert the normalized wind in the tower data binary file to velocities in units of meters per second, use the following equations:

$$U_{tower}(iz, it) = u_{hub} \left(\frac{TI(u)}{1000} U_{tower_norm}(iz, it) + 1 \right),$$

$$V_{tower}(iz, it) = u_{hub} \left(\frac{TI(v)}{1000} V_{tower_norm}(iz, it) \right),$$

and

$$W_{tower}(iz, it) = u_{hub} \left(\frac{TI(w)}{1000} W_{tower_norm}(iz, it) \right).$$

Here TI represents the turbulence intensity as a decimal, not a percentage.

The corresponding lateral locations, Y , and vertical locations, Z , of the tower points are given in units of meters using values of Z_{max} from the file header:

$$Y_{tower} = 0$$
$$Z_{tower}(iz) = Z_{max} - (iz - 1) dz$$