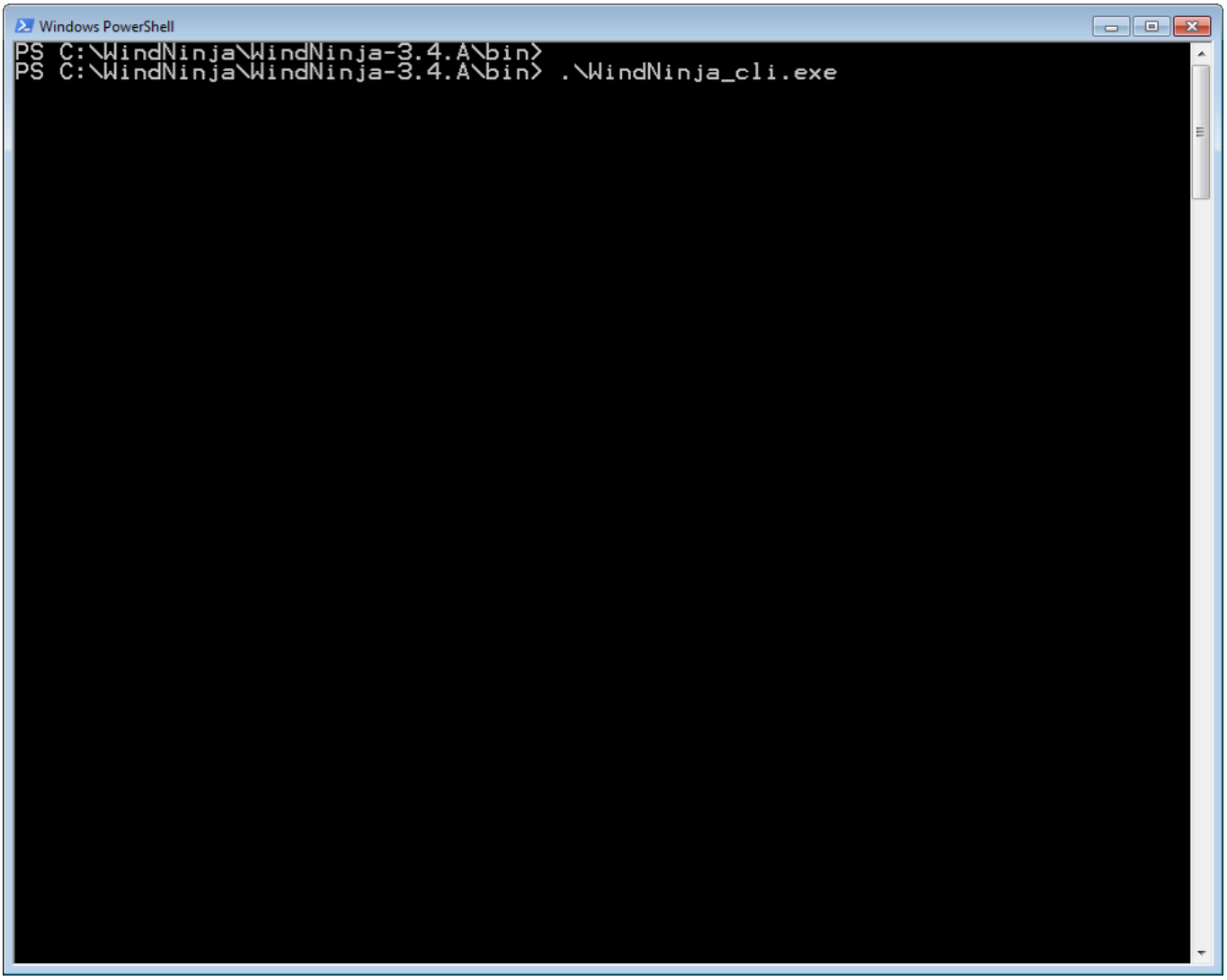
Instructions for using WindNinja's command line interface (CLI)



12/13/2023

Introduction

To allow WindNinja to be used more easily by other programs or through scripting, a command line interface (CLI) has been developed. Programmers should find this very useful, however, most WindNinja users (fire managers and fire modelers) will not use the CLI. This short paper gives a description of how to use the CLI. It assumes that you have some experience running programs from the command line (terminal).

The WindNinja cli is provided as a separate executable called "WindNinja_cli.exe". This executable comes with the normal installation of WindNinja for the Windows operating system (in the "bin" directory). It is also possible to run WindNinja on GNU/Linux. Users interested in the Linux version should visit the WindNinja development website. The CLI executable is dependent on all of the dynamic link libraries provided in the "bin" directory (libcurl.dll, gdal18.dll, etc.) except the Qt libraries "Qt4Core4.dll" and "QtGui4.dll" (used only in the gui version).

Starting A Run

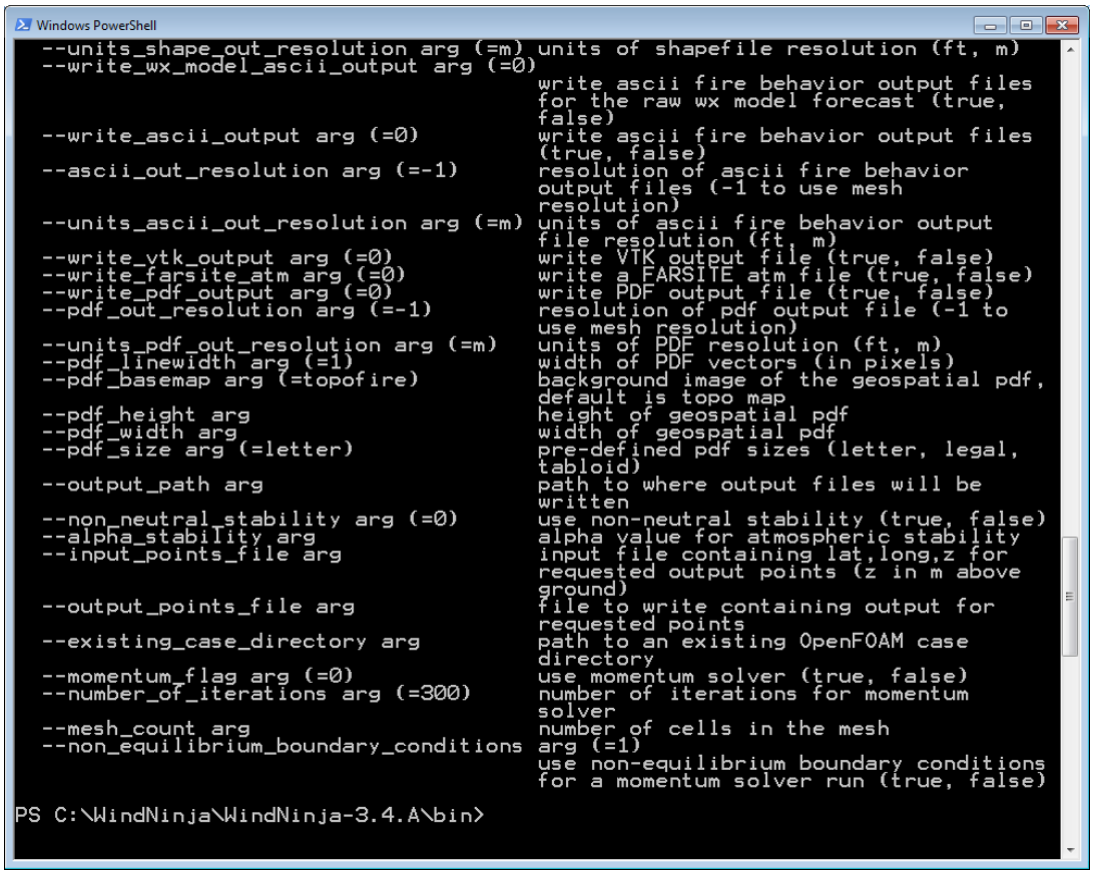
A cli run must be started from a terminal (or "spawned" or something equivalent when called from another program). You type the name of the cli executable ("WindNinja_cli.exe") and then options (arguments) to specify information about the run. The options can either be specified directly from the terminal, by using a configuration file, or some combination of these. To use the terminal, just type the executable followed by the options and associated values. To use a configuration file, just type the name of the executable followed by the name of the configuration file (absolute or relative path from the location of the executable file).

Available Options

The available options with descriptions can be viewed by typing:

```
WindNinja_cli
```

(note that it is optional to add the file extension ".exe" to "WindNinja_cli") A list of the available options should be shown and look similar to this:



Options can be used in any order. Each option has an associated value that can be a string, integer, or float value depending on the option. Some of the options have a default value that is used if the option is not specified. The default value is shown in parenthesis (for example, the `--num_threads` option shown above is defaulted to 1 thread).

Depending on the type of run you are trying to do, certain options are required and some are mutually exclusive (ie. can't both be specified at the same time). If you specify two mutually exclusive options, or don't specify a required option, a message with information on what you did wrong should be shown. There are too many possible combinations of options to describe here, but there is a list of the most basic input requirements for each type of run at the end of this document. Some example configuration files have been included with the installation to show which options to specify for common types of runs. We recommend starting with one of these example configuration files and modifying them for your purpose.

Starting a run using the terminal

The terminal can be used to start a run by simply typing "WindNinja_cli" followed by the option/value pairs separated by a space or "=" like this:

```
WindNinja_cli --num_threads 4 --vegetation=trees ...etc...
```

Starting a run using a configuration file

The configuration files are just text files that list the options and associated values. The example configuration files provided with the normal WindNinja installation are located in the installation's "*etc/windninja/example-files*" directory for a Windows install, or the "*windninja/data*" directory for a Linux build. The files are:

```
cli_domainAverage.cfg

cli_domainAverage_diurnal.cfg

cli_momentumSolver_diurnal.cfg

cli_pointInitialization_diurnal.cfg

cli_wxModelInitialization_diurnal.cfg
```

The filenames give insight into what sort of WindNinja run the configuration file does. For example, the "cli_wxModelInitialization_diurnal.cfg" file does a weather forecast model initialized simulation with diurnal flow turned on. You can open each file to see additional comments describing what the run is doing. The format of a configuration file is as follows:

- "`#`" denotes a comment (to the end of the line)
- Setting an option is like this: `option_name = option_value`
  Note that the "`--`" is not used in the option name in a configuration file.

The contents of a configuration file are shown below:

```
#
# This is an example command line interface (cli) configuration file.
#
# This particular file illustrates the necessary options settings to
# do a weather forecast model initialization run with diurnal winds.
# The weather model is downloaded via the Internet.  The mesh is set
# to a specified resolution of 250 meters.
#
num_threads = 12
elevation_file = C:/XXXX/missoula_valley.tif
initialization_method = wxModelInitialization
time_zone = America/Denver
wx_model_type = NCEP-NAM-12km-SURFACE
forecast_duration = 100
output_wind_height = 20.0
units_output_wind_height = ft
vegetation = trees
diurnal_winds = true
mesh_resolution = 250.0
units_mesh_resolution = m
write_goog_output = true
write_shapefile_output = true
write_ascii_output = true
write_farsite_atm = true
write_wx_model_goog_output = true
write_wx_model_shapefile_output = true
write_wx_model_ascii_output = true
```

To run this particular configuration file, you would just type:

```
WindNinja_cli C:/XXXX/cli_wxModelInitialization_diurnal.cfg
```

where "XXXX" represents the rest of the path to the file.

Starting a run by specifying options from both the terminal and a configuration file

A very useful feature of the WindNinja cli is that you can specify options from both the terminal and a configuration file at the same time. One way to use this feature would be to put the more "general" options in a configuration file, but then specify other more specific options for the run via the terminal. If the same option is specified in both the terminal and the configuration file, the terminal value is used.

As an example of this, you could use the configuration file shown above but "override" the

elevation file, vegetation, and number of threads options by typing this:

```
WindNinja_cli C:/XXXX/cli_wxModelInitialization_diurnal.cfg --elevation_file
C:/XXXX/canyon_fire.asc --vegetation grass --num_threads 4
```

Types of Runs and Required Input Options

This section outlines the 3 different initialization types for WindNinja runs, along with the most basic required inputs for each type of run. These options can be set in a .cfg file, or specified in the terminal as detiled above. Additional options can be viewed by typing:

```
WindNinja_cli
```

Required inputs for ALL runs

Required Inputs:

- `elevation_file`
- `initialization_method`
- `output_wind_height`
- `units_output_wind_height`
- `vegetation`
- Either `mesh_resolution` or `mesh_choice`
- `units_mesh_resolution` - Required if `mesh_resolution` is used

Point Initialization

If the fetch type option is included in the Point Initialization run, additional inputs are required:

- If `fetch_type` = 'Stid' (station ID):
  - `wx_station_filename`
- If `fetch_type` = 'bbox' (bounding box):
  - `fetch_station` - Should be set to true

Domain Average Initialization

Required Inputs:

- `input_speed`
- `input_units`

- `input_direction`
- `input_wind_height`
- `units_input_wind_height`

Weather Model Initialization

Required Inputs:

- `wx_model_type` - Type of the weather model (e.g., 'NCEP-NAM-12km-SURFACE')
- `forecast_duration`