

GNOME

logged in as brian.zelenke | [Logout](#) | [Help/Guide](#) | [About Trac](#) | [Preferences](#)

Home	History	Milestones	Browse Source	View Tickets	New Ticket	Search	Management	Admin
wiki: CommandFiles								

[Start Page](#) | [Index](#) | [History](#)

Last modified 7 weeks ago

GNOME COMMAND.TXT files

GNOME COMMAND.TXT file provide a way to run GNOME in a batch mode. This is used for doing statistical runs (TAP) and for running gnome behind another interface, such as a web site, or other GUI.

These docs are a bit sparse at the moment -- please update as you find new stuff.

Using a COMMAND.TXT File With GNOME

The COMMAND.TXT file allows generation of SPLOT File series to be automated.

A COMMAND file is a text file with GNOME commands. The first line of a COMMAND file must be:

```
[GNOME COMMAND FILE]
```

Primary method:

Launching GNOME with a command file in the GNOME directory.

If there is a COMMAND file called COMMAND.TXT in the same directory as the GNOME executable, when GNOME is launched, GNOME will execute the commands in the file and then quit.

Secondary Method:

Opening/Executing command files from within GNOME

Recent versions of GNOME also support the execution of a command file through the normal "Open" file. If you open a file which is recognized as a GNOME command file, GNOME will execute the commands in the file. Note that GNOME will not automatically quit in this case. (but it will if the quite command is given at the end of the command file)

You can also start GNOME from the command line, and pass the command file name in as an argument:

```
$gnome.exe \c:The\Path\To\The\command_file.txt
```

(note, this is a bit fragile with path handling -- it works best if you pass in the absolute path) We are working on making this approach more robust.

Using an Error Log:

Under typical GNOME usage, any Warning, Error or Note alerts would stop the model and wait for user to hit the OK button. To prevent this behavior, you can specify an Error Log file. If an error log file is specified, the text of the alert will be appended to the error log file. In the case of Warning and Note alerts, the command file continues to execute. In the case of an Error alert, execution of the command file is halted.

```
-- Set the error log file
-----
MESSAGE setField; TO model; errorLog :MyFileName.txt;
```

The commands:

In typical usage, the first part of your command file will setup the data for the scenario(s) you wish to run. There are two ways to do this:

1. use a command to open a GNOME save file
2. use a series of commands of the same type used to setup a location file.

In the second part of the command file, you would typically include one or more lines of the RUNSPILL message which runs a scenario and outputs the LEFiles.

The RUNSPILL message:

Here is an example message which includes all the required parameters:

(NOTE: in the following, long lines are continued to the next line and indented to fit on a page -- in a real file, they must be one long line.)

```
MESSAGE runSpill;TO model; startRelTime 19,07,2000,17,00;
runDurationInHrs 18;timeStepInMinutes 30; numLEs 1000;
```

```
startRelPos 123.7667 W 46.21667 N;outputStepInMinutes 60;
outputFolder :Output17::
```

The options under RUNSPILL are now available in separate parts... this need to be better integrated but should work.

Create spill in command file

```
MESSAGE createSpill; TO model; NAME MC-252; startRelTime 25,06,2010,15,00;
numLEs 1000; startRelPos 91.67 W 28.119198 N;
```

Run the model via the command file without setting a spill.

```
MESSAGE run; TO model; startTime 25,06,2010,15,00; runDurationInHrs 120;
timeStepInMinutes 15; outputStepInMinutes 60;outputFolder :MyOutput::
```

optional parameters : note, outputPath, netcdfPath

Note that for NetCDF output, the NETCDFPATH parameter is given the same way as the OUTPUTPATH parameter.

Initializing a spill from an LE File

A spill can be initialized from an LE File (as exported by GNOME in the "for GNOME Analyst" format"

```
MESSAGE createSpill; TO model; LeFilePath C:\an\example\path\test_splotsFORCST;
```

You can pass in the other relevant parameters, too. I think you can use an LE file for a runSpill message, too.

Optional parameters for the RUNSPILL message:

```
endRelTime, endRelPos, pollutantType, totalMass, massUnits,
windageA, windageB, persistence, z, runBackwards
```

Defaults:

endRelTime	startRelTime
endRelPos	startRelPos
pollutantType	Non-weathering
totalMass	numLEs
massUnits	Barrels
windageA	0.01
windageB	0.04
persistence	0.25
z	0.0 (meters)

pollutantType KEYWORDS:

- CONSERVATIVE or "Non-Weathering"
- BUNKER or "Fuel Oil #6"
- MEDIUMCRUDE or "Medium Crude"
- IFO or "Fuel Oil #4"
- DIESEL
- JP4 or "Kerosene / Jet Fuels"
- GAS or "Gasoline"

When there are two keywords for the same oil, the first is the one used in the MOSS files and the second is the one used in the GNOME interface. You may use either one in the command file.

Note: These key words are NOT case sensitive, but you MUST include any internal spaces.

massUnits KEYWORDS:

BARRELS, GALLONS, CUBICMETERS, KILOGRAMS, METRICTONS, SHORTTONS

MESSAGE SETFIELD

The setField message can be used to change the paramaters of exiting objects. For example, the following properties of a spill can be changed:

numLEs
startRelPos
endRelTime
bWantEndRelTime
endRelPos
bWantEndRelPosition
pollutantType

```
massUnits
totalMass
z
density
ageInHrsWhenReleased
windageA
windageB
persistence
```

example:

```
MESSAGE SETFIELD; TO MC-252; startRelTime 01,07,2010,11,11;
    numLEs 2000; startRelPos 91.11 W 28.1111 N;
    endRelTime 01,07,2010,12,12;
    endRelPos 91.22 W 28.2222 N;
```

MESSAGE createMap

Example:

```
MESSAGE createMap; TO model; TYPE vector; NAME NGulfCoast; PATH ..\coast3.bna;
```

MESSAGE createMover

Examples:

```
MESSAGE createMover; TO Universal Map; TYPE Wind; NAME Offshore;
    PATH winds\OFFSHOREJUNE23PM.txt; speedUnits knots;
```

```
MESSAGE createMover; TO NGulfCoast; TYPE NetCDF; NAME NCOMg;
    PATH currents\NCOMg.nc;
```

Types

```
Wind
NetCDF
CATS
Random
GridCur
PtCur
NetCDFWind
```

Required Parameters

```
NAME NameOfMover
PATH PathToFile
```

Nowcast/forecast model data

```
MESSAGE createMover;TO GOM Map;TYPE NetCDF; NAME GOM.nc;PATH :GOM.nc; MESSAGE createMover;TO Universal
Map;TYPE NetCDF; NAME GOM.nc;PATH :GOM.nc;topFile :GOMTopology.dat; MESSAGE createMap;TO Model;TYPE PtCur;
NAME GOM Map;PATH :GOM.nc;topFile :GOMTopology.dat;
```

The mover can either be added to a vector map or use its own ptcu map. In the latter case the single netcdf file creates both the mover and the map. If it is loaded as a current the map name will attach ' Map'. Topology file path is not necessary for regular grids.

Optional Parameters

```
topFile PathToFile (for a topology file for a netcdf mover)
```

Adding Overlays

Examples

```
MESSAGE OPEN; TO model; PATH ..\Overflights\Overflight_Mobile_2010_0625.txt;
```

To turn off the overlay

```
MESSAGE SETFIELD; TO Overflight_Mobile_2010_0625.txt; bShowOverlay False;
```

Uncertainty

```
MESSAGE setfield;TO model; INCLUDEMINREGRET true;
```

Additional Messages:

```
MESSAGE close; TO model;
MESSAGE clearWinds; TO model;
MESSAGE clearSpills; TO model;
MESSAGE reset; TO model;
MESSAGE quit; TO model;
MESSAGE save; TO model; PATH pathname;
MESSAGE startTimer; TO model;
MESSAGE stopTimer; TO model;
```

Existing command file behavior / workarounds / how to

Work around for Bug::

Wind barbs for winds created via a command file do not show up if you give them a name in the command file.
So... don't use a name when creating a wind mover.

How to set "active" via command file::

(the name you have to use is "bActive")

```
MESSAGE SETFIELD; TO NCOMg; bActive False;
```

An example COMMAND.TXT file

```
[GNOME COMMAND FILE]

-- this would clear all maps etc in case we are not launching GNOME
MESSAGE close; TO model;

-----
-- Set the error log file
-----
MESSAGE setField; TO model; errorLog :MyFileName.txt;

-----
-- Either open a save file
-----
-- MESSAGE open; TO model; PATH yourRelativePathHere;

-----
-- Or use normal messages from wizard commands in location file
-----

MESSAGE setfield;TO model;timeStep 0.10;
-- MESSAGE createMap;TO model;TYPE vector; NAME CR Estuary Map;
  PATH :Columbia River:Crest.bna;
-- MESSAGE createMover;TO CR Estuary Map;TYPE Cats; NAME River Flow.CUR;
  PATH :Columbia River:River Flow.cur;
-- MESSAGE createMover;TO CR Estuary Map;TYPE Cats; NAME CR Tides.CUR;
  PATH :Columbia River:CR Tides.cur
MESSAGE setfield;TO CR Tides.CUR; scaleType constant;
  refP 123.7667 W 46.21667 N; scaleValue -1;
  timeFile :Columbia River:Shio.txt;

MESSAGE createMover;TO Universal Map;TYPE Random; NAME Diffusion;
MESSAGE setfield;TO Diffusion; coefficient 200000;uncertaintyFactor 2
```

```
-----
-- specific setup message not typically used in LocationFiles
-----
MESSAGE clearWinds; TO model;
MESSAGE createMover;TO Universal Map;TYPE Wind; NAME Variable Wind;
  PATH :Columbia River:WindData.wnd; speedUnits knots;
```

```
-----
-- specific messages for the TAP run
-----
MESSAGE runSpill;TO model; startRelTime 19,07,2000,17,00;
  runDurationInHrs 18;timeStepInMinutes 30; numLEs 1000;
  startRelPos 123.7667 W 46.21667 N;outputStepInMinutes 60;
  outputFolder :TapOutput7th:;

MESSAGE runSpill;TO model; pollutantType Kerosene / Jet Fuels;
  totalMass 5000; massUnits GALLONS;startRelTime 19,08,2000,17,00;
  endRelTime 19,08,2000,21,30; runDurationInHrs 18;timeStepInMinutes 30;
  numLEs 1000;startRelPos 123.7667 W 46.21667 N;
  endRelPos 123.7667 W 46.24 N;outputStepInMinutes 60;
  outputFolder :TapOutput8th:;

-----
-- for a hindcasting run
```

```
-----  
MESSAGE runSpill;TO model; runBackwards yes; pollutantType Kerosene / Jet Fuels;  
totalMass 5000; massUnits GALLONS;startRelTime 19,08,2000,17,00;  
endRelTime 19,08,2000,21,30; runDurationInHrs 18;timeStepInMinutes 30;  
numLEs 1000;startRelPos 123.7667 W 46.21667 N;endRelPos 123.7667 W 46.24 N;  
outputStepInMinutes 60;outputFolder :TapOutput8th:;  
-----  
  
-- if we wanted to quit  
--MESSAGE quit; TO model;
```

Additional commands from the GNOME code

► Attachments

Download in other formats:

Plain Text



Powered by [Trac 0.12](#)
By [Edgewall Software](#).

Visit the Trac open source project at
<http://trac.edgewall.org/>

GNOME

logged in as brian.zelenke | [Logout](#) | [Help/Guide](#) | [About Trac](#) | [Preferences](#)

Home	History	Milestones	Browse Source	View Tickets	New Ticket	Search	Management	Admin
----------------------	-------------------------	----------------------------	-------------------------------	------------------------------	----------------------------	------------------------	----------------------------	-----------------------

wiki: [AdditionalCommands](#)

[Start Page](#) | [Index](#) | [History](#)

Last modified 26 hours ago

TClassID setfield parameters

- bActive {t/f} default is true

TMover setfield parameters:

- UncertaintyDuration {val} val>=0
- UncertaintyStartTime {val}

TRandom setfield parameters:

- Coefficient {val} val>0
- coverage {val} old style, same as above
- Uncertaintyfactor {val} val >= 1.0
- DepthDependent {t/f}

TWindMover setfield parameters:

- uncertaintySpeedScale {val}
- uncertaintyAngleScale {val}

TCurrent setfield parameters:

- DownCurUncertainty {val} input positive value, gnome converts
- UpCurUncertainty {val}
- CrossCurUncertainty {val} gnome uses +/- val, or use left right below
- LeftCurUncertainty {val}
- RightCurUncertainty {val}

TCatsmover setfield parameters:

- scaleValue {val}
- EddyDiffusion {val}
- EddyV0 {val}
- scaleType {str}
 - none
 - constant
 - othergrid
- refP {WorldPt}
- timeFile {str}
 - speedUnits {str2} - one or other I believe
 - scaleFactor {val}

TComponentMover Messages

- NAME {str}
- TYPE {str} "Cats" only option for now, requires a PATH
- PATH {str}
- PATTERN {1/2}

setfield parameters

- pat1Angle
- pat2Angle
- pat1Speed
- pat2Speed
- pat1ScaleToValue
- pat2ScaleToValue
- pat1SpeedUnits
- pat2SpeedUnits
- scaleBy {str}
 - windspeed - default
 - windstress
- refP {str} {degLon dirLon degLat dirLat}
- scaleType {str} - "WINDMOVER" the only option for now

TModel Messages:

- RESET
- CLOSE
- CLEARSPILLS
- CLEARWINDS
- OPEN {pathname}
- SAVE {pathname}
- RUNSPILL
- CREATESPILL
- RUN
- STARTTIMER
- STOPTIMER
- QUIT
- CREATEMAP
 - NAME {str}
 - TYPE {str}
 - Vector
 - OSSM
 - PTCUR
 - PATH {str}

Setfield parameters:

- TIMESTEP {hours}
- INCLUDEMINREGRET {t/f} default F
- PREVENTLANDJUMPING {t/f} default T
- MAXDURATION {hours}
- ERRORLOG {path}
- SUPPRESSDRAWING {t/f} default F
- TYPE {str}
 - "dateLimits"
 - StartYear {yearStr} - can be "any" or a year - StartYear {val}
 - StartMonth {val}
 - StartDay {val}
 - EndYear {yearStr} - can be "any" or a year - EndYear {val}
 - EndMonth {val}
 - EndDay {val}

Overlay

- bShowOverlay {t/f}

TMap setfield parameters:

- RefloatHalfLifeInHrs {val}

TMap messages:

- CREATEMOVER
- NAME {str}
- TYPE {str}
 - Random
 - Wind
 - NetCDFWind - needs a path
 - Component - needs a path
 - Cats - needs a path
 - GridCur - needs a path
 - PtCur - needs a path
 - NetCDF - needs a path
 - topFile {str} - optional
 - TideCur - needs a path
- PATH {str}

TVectorMap setfield parameters:

- ExtendedLoLat {val}
- ExtendedHiLat {val}
- ExtendedLoLong {val} HaveMapBoundsLayer()
- ExtendedLeftLong {val}
- ExtendedHiLong {val}
- ExtendedRightLong {val}

LEList

- numLEs
- startRelPos {WorldPoint} allowLLWithoutDir option

- startRelTime {seconds}
- endRelTime - set bWantEndRelTime = true
- bWantEndRelTime {t/f}
- endRelPos - set bWantEndRelPosition = true
- bWantEndRelPosition {t/f}
- pollutantType {str} - optional
 - CONSERVATIVE or Non-Weathering
 - BUNKER or Fuel Oil #6
 - MEDIUMCRUDE or Medium Crude
 - IFO or Fuel Oil #4
 - DIESEL
 - JP4 or Kerosene / Jet Fuels
 - GAS or Gasoline
- massUnits {str} optional
 - BARRELS
 - GALLONS
 - CUBICMETERS
 - KILOGRAMS
 - METRICTONS
 - SHORTTONS
- totalMass
- z
- density
- ageInHrsWhenReleased
- windageA - set for uncertainty too
- windageB
- persistence *persistence of -1 is infinite*
- LEFileReleaseTime

Notes from the Location File Cookbook

- This does not include the scripting options for retrieving data from the dialog boxes

Almost all messages start with the keyword “MESSAGE” and then the instruction key word which determines the type of message, followed by GNOME’s instructions for what to do with the message. Instruction keywords are:

createmap - used to create a map
 createmover - used to create a mover
 setfield - used to set a parameter in the model or a mover

Messages for Model Settings

Timestep

- MESSAGE setfield;TO model;timeStep 0.25;
- MESSAGE setfield; TO model; TYPE dateLimits; startYear any; startMonth July; startDay 1; endYear any; endMonth September; endDay 7;

This sets the limits that we believe are applicable for the dates for the Location File. For example, in Gwydyr Bay, Alaska, we simulate only open water time periods, so we do not think that people should simulate open water conditions in January. This is just a warning. If the date is outside this range, GNOME will put up warning dialog box, but will still allow the user to run their date of choice.

Messages for Maps

GNOME has two types of maps: the Universal maps that covers the entire world, and individual maps that cover only their defined bounds. Movers attached to the Universal Map will be valid throughout the world domain, while movers attached to individual map will only influence LEs (splots) within their map bounds. Since GNOME generally uses a single map, this duality tends to be more of a holdover from OSSM than a functional distinction. The universal map does not need to be created; individual maps do.

- MESSAGE createmap;TO model;NAME <mapName>;TYPE mapType;PATH <pathDescriptor>
- MESSAGE createMap;TO model;TYPE vector; NAME San Diego Map;PATH resnum 10003;

The maps name <mapName> will show up in the left hand list in GNOME; “Universal Map” is the correct title for that map. Valid map types are OSSM and Vector (bna). The PATH is the text resource ID number, usually resnum 10003 (See above example). When referring to these examples to create your own GNOME command file, you can replace the “resnum #####” with the actual path to the file on your computer.

Oftentimes, the map will not extend far enough offshore for the particular application. Be sure to check that there are no Map Bounds in the map file - the two can easily cause conflicts, so use one or the other. GNOME has a special dialog box for extending map boundaries as well as messages for the location files to do the same thing. They four possibilities are:

- ExtendedLoLat moves the southern portion of the map
- ExtendedHiLat " northern "

- ExtendedLoLong moves a side of the map
- ExtendedHiLong moves the other side of the map
- ExtendedLeftLong moves the left side of the map
- ExtendedRightLong moves the right side of the map

Below is an example how to use these commands:

- MESSAGE createmap; TO model; TYPE vector; NAME <mapname>; PATH resnum 10003
- MESSAGE setfield; TO <mapname>; ExtendedLoLat 72.7;

These commands would create the map called <mapname> and extend its southern boundary to 72.7° N. Note that the latitude and longitude values use +/- rather than N/S or E/W. These commands do NOT work with direction descriptors (N,S,E,W).

The refloat half-life of the map can be altered using the following wizard command:

- MESSAGE setfield; TO <mapname>; RefloatHalfLifeInHrs <value>;

Units are in hours and apply to the entire shoreline.

Messages for Movers

Individual movers must be created, assigned to a map and their parameters set. Diffusion is the one type of mover that is always created, so a separate creation message is not needed. All the mover types available in GNOME's diagnostic mode are supported for location files.

All Movers

- MESSAGE createmover;TO <mapName>;TYPE Constant;NAME Offshore;
- MESSAGE createmover;TO <mapName>;TYPE <moverType>;NAME Offshore;[PATH <pathDescriptor>]

SETFIELD values for a Mover

- UncertaintyDuration -- <value> in Hours
- UncertaintyStartTime -- <value> in Hours,

Diffusion

Diffusion is automatically created and assigned to the Universal Map in location files. The diffusivity value (coverage, default 100000) and uncertainty factor (default 2) are only required if values other than the defaults are used.

- MESSAGE setfield;TO diffusion; coverage 500000;uncertaintyFactor 2

-or-

- MESSAGE setfield; TO Diffusion; coefficient 100000; uncertaintyFactor 2;

Note that the eddy diffusivity is termed "coverage" (a holdover from OSSM) and is assumed to be in units of cm²/sec.

In save files for TAP, diffusion is not set as a default. The command to create a diffusion mover is:

- MESSAGE createMover;TO Universal Map; TYPE Random; NAME; Diffusion
- MESSAGE setfield, TO Diffusion; coefficient 100000, uncertaintyFactor 2

Wind moverType:

Constant -- for constant wind Wind -- for variable wind

SETFIELD values for a WindMover

- windage -- <minvalue> <maxValue> percentages as a fraction of 1
- uncertaintySpeedScale -- <value>
- uncertaintyAngleScale -- <value>

MESSAGE setfield;TO Variable Wind;windage 0.02 0.03;UncertaintySpeedScale 1.5;UncertaintyAngleScale .3;UncertaintyDuration 2;UncertaintyStartTime 1;

Currents Requires a path descriptor (e.g. "resnum 10004") CATS SETFIELD values for a CatsMover

- refP -- <longVal> <E or W>< latVal ><N or S>
- scaleType -- "constant","none","othergrid"
- scaleValue -- <theValue>
- EddyDiffusion -- <theValue> in cm²/sec
- ScaleBy -- <WindStress> or <WindSpeed>; WindSpeed is default; used only in component mover.
- DownCurUncertainty -- <theValue> percentage as a fraction of 1
- UpCurUncertainty --<theValue> percentage as a fraction of 1 (if left out, this value will default to - 1*DownCurUncertainty). Must follow on same line as DowCurUncertainty.
- CrossCurUncertainty -- <theValue> percentage as a fraction of 1 (sets both right and left cross current uncertainty)

- LeftCurUncertainty --<theValue> percentage as a fraction of 1
- RightCurUncertainty--<theValue> percentage as a fraction of 1 (if left out, this value defaults to LeftCurUncertainty). Must follow on same line as LeftCurUncertainty.

Examples for Save File Commands for CATS current files are shown below.

--Constant pattern

- MESSAGE createMover;TO ROPME Sea Area Map;TYPE Cats; NAME Reverse Estuary;PATH resnum 10004;
- MESSAGE setfield;TO Reverse Estuary; scaleType constant;refP 56.23757 E 26.574903 N; scaleValue 0.2;

--Tides - with SHIO coefficients

- MESSAGE createMover;TO San Diego Map;TYPE Cats; NAME tides.cur;PATH resnum 10004;
- MESSAGE setfield;TO tides.cur; scaleType constant;refP 117.2300 W 32.681667 N; scaleValue 1; timeFile resNum 10005; EddyDiffusion 10000;

--Tides - with shio tide heights file (gnome takes d/dt)

- MESSAGE createMover;TO San Juan Map;TYPE Cats; NAME Ebb tides.cur;PATH resnum 10004;
- MESSAGE setfield;TO Ebb tides.cur; scaleType constant;refP 66.12833333 W 18.47133333 N; scaleValue 1; timeFile resNum 10005; scaleValue 0.15;
- MESSAGE setfield;TO Ebb tides.cur; EddyDiffusion 100000; EddyV0 0.1;

Note: the speed units will end up in knots, so plan the scale factor accordingly. Seems that the scaleValue isn't applied.

--Tides - with tidal currents time series

- MESSAGE createMover;TO San Juan Map;TYPE Cats; NAME Ebb tides.cur;PATH resnum 10004;
- MESSAGE setfield;TO Ebb tides.cur; scaleType constant;refP 66.12833333 W 18.47133333 N; scaleValue 1; timeFile resNum 10005; speedUnits knots;
- MESSAGE setfield;TO Ebb tides.cur; EddyDiffusion 100000; EddyV0 0.1;

Note: speedUnits can be: knots,mps,mph,meterspersec, milesperhour.

--Component, one pattern

- MESSAGE createMover;TO ROPME Sea Area Map;TYPE Component; NAME Wind Driven Currents;
- MESSAGE createMover;TO Wind Driven Currents;TYPE Cats; PATTERN 1; NAME NNW10;PATH resnum 10006;
- MESSAGE setfield;TO Wind Driven Currents; scaleType WINDMOVER;refP 49.744965 E 27.285172 N;
- MESSAGE setfield;TO Wind Driven Currents; pat1Angle 337.5; pat1Speed 10; pat1SpeedUnits knots; pat1ScaleToValue 0.15;

--Component, two patterns

- MESSAGE createMover;TO San Diego Map;TYPE Component; NAME Wind Driven Currents;
- MESSAGE createMover;TO Wind Driven Currents;TYPE Cats; PATTERN 1; NAME NNWat15kt.cur;PATH resnum 10007;
- MESSAGE setfield;TO Wind Driven Currents; scaleType WINDMOVER;refP 117.1295 W 32.644333 N;
- MESSAGE setfield;TO Wind Driven Currents; pat1Angle 337.5; pat1Speed 15; pat1SpeedUnits knots; pat1ScaleToValue 0.03;
- MESSAGE createMover;TO Wind Driven Currents;TYPE Cats; PATTERN 2; NAME WSWat15kt.cur;PATH resnum 10008;
- MESSAGE setfield;TO Wind Driven Currents; scaleType WINDMOVER;refP 117.1295 W 32.644333 ;
- MESSAGE setfield;TO Wind Driven Currents; pat2Angle 247.5; pat2Speed 15; pat2SpeedUnits knots; pat2ScaleToValue 0.03;
- MESSAGE setfield;TO Wind Driven Currents; scaleBy windstress;

Window Messages

[WMSG <dialogResNum>] -- messages to be sent after dialog is complete

- MESSAGE setfield;TO <objName>;<fieldName> <valueString>;
- 8MESSAGE setfield;TO <objName>;<fieldName> \$1;

<fieldName> are words like "scale","active"

Zoom Limit Limits of user zoom of display in Standard mode. - limits how close a user can zoom in (no limit in Diagnostic mode). minkilometersperinch 5; Note that this is only enabled in Standard mode, so for debugging purposes, you can't use Diagnostic Mode.

Partial paths - still supports old style Mac : delimiters, also Unix \ and Windows / . Could be problems if file name contains any of these characters (Mac does allow this, Windows does not). will first look relative to command file and put output relative to command file, but will try to resolve to application folder, save file folder or working directory before giving up. Command line paths - gnome will look for command file relative to the working directory Full paths should work in platform appropriate style

Download in other formats:

Plain Text



Powered by [Trac 0.12](#)
By [Edgewall Software](#).

Visit the Trac open source project at
<http://trac.edgewall.org/>