

# Package ‘Reanalysis’

May 22, 2017

**Type** Package

**Title** Reads reanalysis data for CRHM

**Version** 1.0.3

**Date** 2017-03-28

**Author** Kevin Shook, Centre for Hydrology, University of Saskatchewan

**Maintainer** Kevin Shook <kevin.shook@usask.ca>

**Description** Functions to read and process files of reanalysis data from WATCH, ERA and NARR.

**Depends** R (>= 3.1)

**Imports** CRHMr (>= 2.2.9), lubridate(>= 1.3), timeDate, stringr(>= 1.0), zoo, proj4, RNetCDF, reshape2

**License** GPL-3

**LazyData** true

**URL** [www.usask.ca/hydrology](http://www.usask.ca/hydrology)

**RoxygenNote** 6.0.1

**NeedsCompilation** no

## R topics documented:

|   |    |
|---|----|
| Reanalysis-package . . . . .                | 2  |
| ERAdailyArealPrecip . . . . .               | 2  |
| ERAdaccum . . . . .                         | 3  |
| ERAggetArealPrecip . . . . .                | 4  |
| ERAggetMultipleLocationTimeseries . . . . . | 5  |
| ERAggetNearestTimeseries . . . . .          | 7  |
| ERAhourlyAirtemp . . . . .                  | 9  |
| ERAhourlyLongwave . . . . .                 | 10 |
| ERAhourlyPrecip . . . . .                   | 11 |
| ERAhourlyShortwave . . . . .                | 12 |
| ERAhourlyVP . . . . .                       | 13 |
| ERAhourlyWindspeed . . . . .                | 14 |
| land . . . . .                              | 15 |

|                                     |           |
|-------------------------------------|-----------|
| NARRdownloadNetCDF . . . . .        | 16        |
| NARRgetNearestTimeseries . . . . .  | 17        |
| WATCHcreateHourlyWFDEIobs . . . . . | 18        |
| WATCHcreateHourlyWFDobs . . . . .   | 20        |
| WATCHcreateWFDEIobs . . . . .       | 22        |
| WATCHcreateWFDobs . . . . .         | 23        |
| WATCHdailyArealPrecip . . . . .     | 24        |
| WATCHgetWFDarealPrecip . . . . .    | 25        |
| WATCHgroupWFDEIobs . . . . .        | 26        |
| WATCHgroupWFDobs . . . . .          | 28        |
| WATCHhourlyObs . . . . .            | 29        |
| <b>Index</b>                        | <b>31</b> |

---

|                    |   |
|--------------------|---|
| Reanalysis-package | <i>Contains functions to download and process reanalysis data</i> |
|--------------------|---|

---

**Description**

The package contains functions to download and process reanalysis data. Functions are provided for ETA, NARR and WATCH datasets. Because there are so many functions and all of the datasets are slightly different, each function has the prefix of its dataset.

**References**

To cite **Reanalysis** in publications, use the command `citation('Reanalysis')` to get the current version of the citation.

The CRHM program is described in:  
*Pomeroy, John W, D M Gray, T Brown, N Hedstrom, W L Quinton, R J Granger, and S K Carey. 2007. "The Cold Regions Hydrological Model : A Platform for Basing Process Representation and Model Structure on Physical Evidence". Hydrological Processes 21 (19): 2650-2567.*

The CRHM model may be downloaded from <http://www.usask.ca/hydrology/CRHM.php>.

---

|                     |   |
|---------------------|---|
| ERAdailyArealPrecip | <i>Calculates daily ERA areal precipitation</i> |
|---------------------|---|

---

**Description**

Calculates the daily total ERA precipitation, for all locations.

**Usage**

ERAdailyArealPrecip(ERAarealPrecip)

**Arguments**

ERAarealPrecip Required. The three-hour ERA areal precipitation as returned by ERAgetArealPrecip.

**Value**

If unsuccessful, returns FALSE. If successful, returns a list containing the following:

| <b>Name</b>  | <b>meaning</b>                             |
|--------------|--|
| daily_precip | 3d array (lon x lat x date) of precip (mm) |
| lonres       | longitude resolution (°)                   |
| latres       | latitude resolution (°)                    |
| minLon       | minimum longitude of data(°W)              |
| maxLon       | maximum longitude of data(°W)              |
| minLat       | minimum latitude of data(°)                |
| maxLat       | maximum latitude of data(°)                |
| date         | the date of each layer                     |

**Author(s)**

Kevin Shook

**Examples**

```
## Not run: daily_precip <- ERAdailyArealPrecip(threehour_precip)
```

---

ERAdeaccum

*Deaccumulates ERA cumulative time series*

---

**Description**

This function is used to deaccumulate variables stored as 12-hour cumulative values by ERA to 3-hour values. It is called by other functions, but may also be useful as a stand-alone functions. Note that this function only works for a single location, i.e. *NOT* areal values.

**Usage**

```
ERAdeaccum(ERAobs, colnum = 1, quiet = TRUE, logfile = "")
```

**Arguments**

|         |  |
|---------|--|
| ERAobs  | Required. A <b>CRHMr</b> obs dataframe of ERA data, created by ERAgetnearestTimeSeries.  |
| colnum  | Optional. The column number containing the values to be deaccumulated, not including the datetime. Default is column 1.  |
| quiet   | Optional. Suppresses display of messages, except for errors. If you are calling this function in an R script, you will usually leave quiet=TRUE (i.e. the default). If you are working interactively, you will probably want to set quiet=FALSE. |
| logfile | Optional. Name of the file to be used for logging the action. Normally not used  |

**Value**

If successful, returns an obs dataframe containing the deaccumulated value. If unsuccessful, returns the value FALSE.

**Author(s)**

Kevin Shook

**Examples**

```
## Not run:
deaccum <- ERAdeaccum(ERAobs)
## End(Not run)
```

---

|                   |                                    |
|-------------------|------------------------------------|
| ERAgetArealPrecip | <i>Get ERA areal precipitation</i> |
|-------------------|------------------------------------|

---

**Description**

Extracts the areal precipitation for all time intervals from the ERA NetCDF file for a region. If the data are 3-hour values accumulated over 12-hour periods, then they will be deaccumulated, and negative precipitation will be set to zero

**Usage**

```
ERAgetArealPrecip(ncdfFile, timezone = "", quiet = TRUE)
```

**Arguments**

|          |  |
|----------|--|
| ncdfFile | Required. Name of the NetCDF file containing ERA data.   |
| timezone | Required. The name of the timezone of the data as a character string. This should be the timezone of your data, but omitting daylight savings time. Note that the timezone code is specific to your OS. To avoid problems, you should use a timezone without daylight savings time. Under Linux, you can use 'CST' and 'MST' for Central Standard or Mountain Standard time, respectively. Under Windows or OSX, you can use 'etc/GMT+6' or 'etc/GMT+7' for Central Standard and Mountain Standard time. DO NOT use 'America/Regina' as the time zone, as it includes historical changes between standard and daylight savings time. |
| quiet    | Optional. Suppresses display of messages, except for errors. If you are calling this function in an R script, you will usually leave quiet=TRUE (i.e. the default). If you are working interactively, you will probably want to set quiet=FALSE  |

**Value**

If unsuccessful, returns FALSE. If successful, returns a list containing the following:

| Name     | meaning                                    |
|----------|--|
| precip   | 3d array (lon x lat x time) of precip (mm) |
| lonres   | longitude resolution (°)                   |
| latres   | latitude resolution (°)                    |
| minLon   | mininum longitude of data(°W)              |
| maxLon   | maximum longitude of data(°W)              |
| minLat   | minimum latitude of data(°)                |
| maxLat   | maximum latitude of data(°)                |
| datetime | the datetime of each time layer            |

### Examples

```
## Not run: threehour_precip <-
ERAgetArealPrecip('_grib2netcdf-atls17-95e2cf679cd58ee9b4db4dd119a05a8d-szSVRK.nc',
  timezone='CST')

## End(Not run)
```

---

ERAgetMultipleLocationTimeseries

*Extracts timeseries of ERA data nearest to multiple specified locations*

---

### Description

Reads a NetCDF file containing ERA reanalysis data and extracts the timeseries of the specified variable for each location. This is faster than calling the function ERAgetNearestTimeseries for each location, as the reanalysis data are only read in once. Some of the the commonly-used variables are:

| Parameter                           | Units            | Variable |
|-------------------------------------|------------------|----------|
| 10 m eastward wind component        | m/s              | u10      |
| 10 m northward wind component       | m/s              | v10      |
| 2 metre temperature                 | K                | t2m      |
| 2 metre dewpoint                    | K                | d2m      |
| Downward surface solar radiation*   | J/m <sup>2</sup> | ssrd     |
| Downward surface thermal radiation* | J/m <sup>2</sup> | strd     |
| Surface net solar radiation*        | J/m <sup>2</sup> | ssr      |
| Surface net thermal radiation*      | J/m <sup>2</sup> | str      |
| Total precipitation*                | m of water       | tp       |

Parameters marked with an asterisk are *cumulative* values, and must be deaccumulated using the deaccumERA function.

**Usage**

```
ERAgetMultipleLocationTimeseries(ncdfFile, varName, siteNames, outDir = "",
  pointLons, pointLats, projection = "+proj=utm +zone=13 +ellps=WGS84",
  timezones = "", quiet = TRUE, logfile = "")
```

**Arguments**

|            |   |
|------------|---|
| ncdfFile   | Required. Name of the NetCDF file containing ERA data.  |
| varName    | Required. Name of the NetCDF variable to extract.   |
| siteNames  | Required. A vector containing the names of the sites. The names will be used for the output obs files.  |
| outDir     | Optional. Directory to hold output files. If not specified, the current directory will be used.   |
| pointLons  | Required. A vector containing decimal longitudes of desired locations. Note that the NetCDF longitude is 0-360°, so add 360 to negative longitudes.   |
| pointLats  | Required. A vector containing decimal latitudes of desired location.  |
| projection | Optional. Projection to be used to convert latitudes and longitudes to locations. Used for finding the nearest ERA gridpoint. The default, '+proj=utm +zone=13 +ellps=WGS84', is <i>only</i> valid for Western Canada. If you are processing data for the whole world, you can use the August Epicycloidal Projection which is '+proj=august +lon_0=90w'.   |
| timezones  | Required. A vector containing the name of the timezone of the data as a character string. This should be the timezone of your data, but omitting daylight savings time. Note that the timezone code is specific to your OS. To avoid problems, you should use a timezone without daylight savings time. Under Linux, you can use 'CST' and 'MST' for Central Standard or Mountain Standard time, respectively. Under Windows or OSX, you can use 'etc/GMT+6' or 'etc/GMT+7' for Central Standard and Mountain Standard time. DO NOT use 'America/Regina' as the time zone, as it includes historical changes between standard and daylight savings time. If there are fewer timezone values than variable names, the timezone will be recycled. |
| quiet      | Optional. Suppresses display of messages, except for errors. If you are calling this function in an R script, you will usually leave quiet=TRUE (i.e. the default). If you are working interactively, you will probably want to set quiet=FALSE, as it will list the actual latitude and longitude of the ERA tile.   |
| logfile    | Optional. Name of the file to be used for logging the action. Normally not used.  |

**Value**

If unsuccessful, returns FALSE. If successful, returns TRUE and all variables for all locations are written to obs files in the specified directory.

**Author(s)**

Kevin Shook

**See Also**

[ERAgetNearestTimeseries ERAdeaccum](#)

**Examples**

```
## Not run:
ERAgetMultipleLocationTimeseries('ssrd.nc', 'ssrd', s$`Station Names`,
pointLons = s$Longitude, pointLats = s$Latitude, projection='+proj=august +lon_0=90w',
timezones=s$timezone, quiet=FALSE)
## End(Not run)
```

---

ERAgetNearestTimeseries

*Finds timeseries of ERA data nearest to specified location*

---

**Description**

Reads a NetCDF file containing ERA reanalysis data and extracts the timeseries of the specified variable. Some of the the commonly-used variables are:

| Parameter                           | Units            | Variable |
|-------------------------------------|------------------|----------|
| 10 m eastward wind component        | m/s              | u10      |
| 10 m northward wind component       | m/s              | v10      |
| 2 metre temperature                 | K                | t2m      |
| 2 metre dewpoint                    | K                | d2m      |
| Downward surface solar radiation*   | J/m <sup>2</sup> | ssrd     |
| Downward surface thermal radiation* | J/m <sup>2</sup> | strd     |
| Surface net solar radiation*        | J/m <sup>2</sup> | ssr      |
| Surface net thermal radiation*      | J/m <sup>2</sup> | str      |
| Total precipitation*                | m of water       | tp       |

Parameters marked with an asterisk are *cumulative* values, and must be deaccumulated using the deaccumERA function.

**Usage**

```
ERAgetNearestTimeseries(ncdfFile, varName, pointLon, pointLat,
  projection = "+proj=utm +zone=13 +ellps=WGS84", timezone = "",
  quiet = TRUE, logfile = "")
```

**Arguments**

|          |   |
|----------|---|
| ncdfFile | Required. Name of the NetCDF file containing ERA data.  |
| varName  | Required. Name of the NetCDF variable to extract.   |
| pointLon | Required. Decimal longitude of desired location. Note that the NetCDF longitude is 0-360°, so add 360 to negative longitudes. |

|            |  |
|------------|--|
| pointLat   | Required. Decimal latitude of desired location.  |
| projection | Optional. Projection to be used to convert latitudes and longitudes to locations. Used for finding the nearest ERA gridpoint. The default, '+proj=utm +zone=13 +ellps=WGS84', is <i>only</i> valid for Western Canada. If you are processing data for the whole world, you can use the August Epicycloidal Projection which is '+proj=august +lon_0=90w'.  |
| timezone   | Required. The name of the timezone of the data as a character string. This should be the timezone of your data, but omitting daylight savings time. Note that the timezone code is specific to your OS. To avoid problems, you should use a timezone without daylight savings time. Under Linux, you can use 'CST' and 'MST' for Central Standard or Mountain Standard time, respectively. Under Windows or OSX, you can use 'etc/GMT+6' or 'etc/GMT+7' for Central Standard and Mountain Standard time. DO NOT use 'America/Regina' as the time zone, as it includes historical changes between standard and daylight savings time. |
| quiet      | Optional. Suppresses display of messages, except for errors. If you are calling this function in an R script, you will usually leave quiet=TRUE (i.e. the default). If you are working interactively, you will probably want to set quiet=FALSE, as it will list the actual latitude and longitude of the ERA tile.  |
| logfile    | Optional. Name of the file to be used for logging the action. Normally not used.   |

### Value

If unsuccessful, returns FALSE. If successful, returns a standard **CRHMr** dataframe containing the datetime and the extracted data, which are unpacked (i.e. the NetCDF multiplier and offset have been applied).

### Author(s)

Kevin Shook

### See Also

[ERAdeaccum](#)

### Examples

```
## Not run:
ssrd <- ERAgetNearestTimeseries('ssrd.nc', 'ssrd', 243.5, 51.69, timezone='Etc/GMT+7')
t2m <- ERAgetNearestTimeseries('t2m.nc', 't2m', 243.5, 51.69, timezone='Etc/GMT+7')
tp <- ERAgetNearestTimeseries('tp.nc', 'tp', 243.5, 51.69, timezone='Etc/GMT+7')
strd <- ERAgetNearestTimeseries('strd.nc', 'strd', 243.5, 51.69, timezone='Etc/GMT+7')
u10 <- ERAgetNearestTimeseries('u10v10.nc', 'u10', 243.5, 51.69, timezone='Etc/GMT+7')
v10 <- ERAgetNearestTimeseries('u10v10.nc', 'v10', 243.5, 51.69, timezone='Etc/GMT+7')
d2m <- ERAgetNearestTimeseries('d2m.nc', 'd2m', 243.5, 51.69, timezone='Etc/GMT+7')
## End(Not run)
```



---

|                  |   |
|------------------|---|
| ERAhourlyAirtemp | <i>Estimates hourly air temperatures from ERA 3 hourly values</i> |
|------------------|---|

---

## Description

Interpolates ERA 3-hour instantaneous air temperatures to hourly values. The ERA air temperatures are first converted from K to °C, if required.

## Usage

```
ERAhourlyAirtemp(ERAt2m, t2mColnum = 1, method = "linear", quiet = TRUE,
  logfile = "")
```

## Arguments

|           |  |
|-----------|--|
| ERAt2m    | Required. The <b>CRHMr</b> obs dataframe of ERA t2m values. The values must not be deaccumulated, as the deaccumERA function is called by this function.   |
| t2mColnum | Optional. The column number containing the t2m values, not including the datetime. Default is column 1.  |
| method    | Optional. The methods to be used for interpolation of the air temperature. Currently supported methods are 'linear' and 'spline'. The default is to use linear interpolation.  |
| quiet     | Optional. Suppresses display of messages, except for errors. If you are calling this function in an R script, you will usually leave quiet=TRUE (i.e. the default). If you are working interactively, you will probably want to set quiet=FALSE. |
| logfile   | Optional. Name of the file to be used for logging the action. Normally not used.   |

## Value

If successful, returns an obs dataframe containing the interpolated hourly 2m air temperatures. If unsuccessful, returns the value FALSE.

## Author(s)

Kevin Shook

## See Also

[ERAGetNearestTimeseries interpolate](#)

## Examples

```
## Not run:
hourlyTemps <- ERAhourlyAirtemp(t2m)
## End(Not run)
```

---

|                   |  |
|-------------------|--|
| ERAhourlyLongwave | <i>Estimates hourly incoming longwave radiation from ERA 3 hourly values</i> |
|-------------------|--|

---

## Description

This function is called *after* using the function `getNearestERAtimeseries`. This function deaccumulates the 12-hour cumulative values, using the function `ERAdeaccum` and interpolates the 3-hour values to hourly values, based on the extra-terrestrial hourly radiation.

## Usage

```
ERAhourlyLongwave(ERASTrd, strdColnum = 1, ERAt2m, t2mColnum = 1,
  method = "linear", quiet = TRUE, logfile = "")
```

## Arguments

|            |   |
|------------|---|
| ERASTrd    | Required. The <b>CRHMr</b> obs dataframe of ERA strd values. The values must not be deaccumulated, as the <code>ERAdeaccum</code> function is called by this function.  |
| strdColnum | Optional. The column number containing the strd values, not including the date-time. Default is column 1.   |
| ERAt2m     | Required. The <b>CRHMr</b> obs dataframe of ERA t2m values. The values must not be deaccumulated, as the <code>deaccumERA</code> function is called by this function.   |
| t2mColnum  | Optional. The column number containing the t2m values, not including the datetime. Default is column 1.   |
| method     | Optional. The methods to be used for interpolation of the air temperature. Currently supported methods are 'linear' and 'spline'. The default is to use linear interpolation.   |
| quiet      | Optional. Suppresses display of messages, except for errors. If you are calling this function in an R script, you will usually leave <code>quiet=TRUE</code> (i.e. the default). If you are working interactively, you will probably want to set <code>quiet=FALSE</code> . |
| logfile    | Optional. Name of the file to be used for logging the action. Normally not used   |

## Value

If successful, returns an obs dataframe containing the interpolated hourly ERA incoming longwave radiation ( $Q_{li}$ ) in  $W/m^2$ , and the hourly 2m air temperatures ( $t$ ) in  $^{\circ}C$ . If unsuccessful, returns the value `FALSE`.

## Author(s)

Kevin Shook

## References

The value of the Stefan-Boltzmann constant was obtained from <http://physics.nist.gov/cgi-bin/cuu/Value?sigma>

**See Also**

[ERAgetCodeNearestTimeseries](#) [ERAdeaccum](#) [ERAhourlyShortwave](#)

**Examples**

```
## Not run:
hourlyQli <- ERAhourlyLongwave(ERAstrd=strd, ERA2m=t2m)
## End(Not run)
```

---

|                 |   |
|-----------------|---|
| ERAhourlyPrecip | <i>Estimates hourly precipitation from ERA cumulative 3 hourly values</i> |
|-----------------|---|

---

**Description**

This function is called *after* using the function `getNearestERAtimeseries`. This function removes negative values, deaccumulates the 12-hour cumulative values to 3-hour values, using the function `deaccumERA`, and divides the results by 3 to give to hourly values.

**Usage**

```
ERAhourlyPrecip(ERAtp, tpColnum = 1, quiet = TRUE, logfile = "")
```

**Arguments**

|          |   |
|----------|---|
| ERAtp    | Required. The <b>CRHMr</b> obs dataframe of ERA tp values. The values must not be deaccumulated, as the <code>deaccumERA</code> function is called by this function.  |
| tpColnum | Optional. The column number containing the tp values, not including the date-time. Default is column 1.   |
| quiet    | Optional. Suppresses display of messages, except for errors. If you are calling this function in an R script, you will usually leave <code>quiet=TRUE</code> (i.e. the default). If you are working interactively, you will probably want to set <code>quiet=FALSE</code> . |
| logfile  | Optional. Name of the file to be used for logging the action. Normally not used   |

**Value**

If successful, returns an obs dataframe containing the interpolated hourly precipitation (p). If unsuccessful, returns the value `FALSE`.

**Author(s)**

Kevin Shook

**See Also**

[ERAgetCodeNearestTimeseries](#)

## Examples

```
## Not run:
hourlyP <- ERAhourlyPrecip(tp)
## End(Not run)
```

---

|                    |   |
|--------------------|---|
| ERAhourlyShortwave | <i>Estimates hourly incoming shortwave radiation from ERA 3 hourly values</i> |
|--------------------|---|

---

## Description

This function is called *after* using the function `getNearestERAtimeseries`. This function removes negative values, deaccumulates the 12-hour cumulative values, using the function `ERAdaccum` and interpolates the 3-hour values to hourly values, based on the extra-terrestrial hourly radiation.

## Usage

```
ERAhourlyShortwave(ERAssrd, ssrdColnum = 1, latitude, sunTimeOffset = 2,
  solarMethod = "simpleMaxSolar", quiet = TRUE, logfile = "")
```

## Arguments

|               |   |
|---------------|---|
| ERAssrd       | Required. The <b>CRHMr</b> obs dataframe of ERA ssrd values. The values must not be deaccumulated, as the <code>deaccumERA</code> function is called by this function.  |
| ssrdColnum    | Optional. The column number containing the ssrd values, not including the datetime. Default is column 1.  |
| latitude      | Required. The latitude of the point at which the ssrd data is extracted. This value is used to calculate the extra-terrestrial incoming solar radiation.  |
| sunTimeOffset | Optional. Number of hours that local noon is offset from solar noon. The default is 2 hours.  |
| solarMethod   | Optional. The method to be used for calculating the extra-terrestrial radiation. Default is 'simpleMaxSolar'.   |
| quiet         | Optional. Suppresses display of messages, except for errors. If you are calling this function in an R script, you will usually leave <code>quiet=TRUE</code> (i.e. the default). If you are working interactively, you will probably want to set <code>quiet=FALSE</code> . |
| logfile       | Optional. Name of the file to be used for logging the action. Normally not used   |

## Value

If successful, returns an obs dataframe containing the interpolated hourly ERA incoming shortwave radiation (Qsi). If unsuccessful, returns the value FALSE.

## Author(s)

Kevin Shook

**See Also**

[ERAgetNearestTimeseries ERAhourlyLongwave distributeQsi](#)

**Examples**

```
## Not run:
hourlyQsi <- ERAhourlyShortwave(ssrd, latitude = 51.69)
## End(Not run)
```

---

|             |  |
|-------------|--|
| ERAhourlyVP | <i>Calculates the hourly vapour pressure from the 3-hour ERA dew point temperature</i> |
|-------------|--|

---

**Description**

Interpolates ERA 3-hour instantaneous dew point temperatures to hourly values. The ERA dew point temperatures are first converted from K to °C, if required.

**Usage**

```
ERAhourlyVP(ERAd2m, d2mColnum = 1, method = "linear", quiet = TRUE,
  logfile = "")
```

**Arguments**

|           |  |
|-----------|--|
| ERAd2m    | Required. The <b>CRHMR</b> obs dataframe of ERA d2m values. The values must not be deaccumulated, as the deaccumERA function is called by this function.   |
| d2mColnum | Optional. The column number containing the d2m values, not including the datetime. Default is column 1.  |
| method    | Optional. The methods to be used for interpolation of the dew point temperature. Currently supported methods are ‘linear’ and ‘spline’. The default is to use linear interpolation.  |
| quiet     | Optional. Suppresses display of messages, except for errors. If you are calling this function in an R script, you will usually leave quiet=TRUE (i.e. the default). If you are working interactively, you will probably want to set quiet=FALSE. |
| logfile   | Optional. Name of the file to be used for logging the action. Normally not used.   |

**Value**

If successful, returns an obs dataframe containing the interpolated hourly 2m vapour pressures (ea) in kPa. If unsuccessful, returns the value FALSE.

**Author(s)**

Kevin Shook

**See Also**

[ERAgetNearestTimeseries interpolate ERAhourlyAirtemp](#)

**Examples**

```
## Not run:
hourlyVP <- ERAhourlyVP(d2m)
## End(Not run)
```

---

|                    |  |
|--------------------|--|
| ERAhourlyWindspeed | <i>Estimates hourly wind speeds from ERA 3 hourly wind vectors</i> |
|--------------------|--|

---

**Description**

Estimates hourly wind speeds from ERA 3 hourly wind vectors

**Usage**

```
ERAhourlyWindspeed(ERAu10, u10Colnum = 1, ERAv10, v10Colnum = 1,
  method = "linear", quiet = TRUE, logfile = "")
```

**Arguments**

|           |  |
|-----------|--|
| ERAu10    | Required. The <b>CRHMR</b> obs dataframe of ERA u10 wind vector values. The values must not be deaccumulated, as the deaccumERA function is called by this function.   |
| u10Colnum | Optional. The column number containing the u10 values, not including the date-time. Default is column 1.   |
| ERAv10    | Required. The <b>CRHMR</b> obs dataframe of ERA v10 wind vector values. The values must not be deaccumulated, as the deaccumERA function is vcalled by this function.  |
| v10Colnum | Optional. The column number containing the v10 values, not including the date-time. Default is column 1.   |
| method    | Optional. The methods to be used for interpolation of the wind speeds. Currently supported methods are 'linear' and 'spline'. The default is to use linear interpolation.  |
| quiet     | Optional. Suppresses display of messages, except for errors. If you are calling this function in an R script, you will usually leave quiet=TRUE (i.e. the default). If you are working interactively, you will probably want to set quiet=FALSE. |
| logfile   | Optional. Name of the file to be used for logging the action. Normally not used.   |

**Value**

If successful, returns an obs dataframe containing the interpolated hourly 10m wind speeds (u10) in m/s and directions in degrees. If unsuccessful, returns the value FALSE.

**Author(s)**

Kevin Shook

**See Also**[ERAGetNearestTimeseries interpolate](#)**Examples**

```
## Not run:  
hourlyU <- ERAhourlyWindspeed(ERAU10=u10, ERAv10=v10)  
## End(Not run)
```

---

|      |                       |
|------|-----------------------|
| land | <i>Land locations</i> |
|------|-----------------------|

---

**Description**

A dataframe containing the information about the locations inside the WFD netCDF files. This data is used by the function [WATCHcreateHourlyWFDobs](#).

**Usage**

land

**Format**

A dataframe with 67420 rows and 6 variables:

**Land** number of the grid location

**Longitude** longitude of the centre grid square

**Latitude** latitude of the centre grid square

**Ht.m** elevation of the centre grid square

**glon** x-location of the grid square

**glat** y-location of the grid square

**Source**

This data was extracted from a file distributed with the WFD data, which can be downloaded from <http://www.eu-watch.org/>.

---

|                    |                                    |
|--------------------|------------------------------------|
| NARRdownloadNetCDF | <i>Downloads NARR NetCDF files</i> |
|--------------------|------------------------------------|

---

## Description

Downloads NARR NetCDF files for a specified variable for a specified time step and range of years.

## Usage

```
NARRdownloadNetCDF(interval = "daily", startYear = 1979, endYear = 1979,
  destination = ".", variable = "precip", quiet = FALSE)
```

## Arguments

|             |  |
|-------------|--|
| interval    | Optional. Time interval of NARR data. Can be '3h' or 'daily', which is the default.  |
| startYear   | Optional. The first year to download. Default is 1979.   |
| endYear     | Optional. The last year to download. Default is 1979.  |
| destination | Optional. The destination directory for the downloaded files. The default is the current directory.  |
| variable    | Optional. The variable to be downloaded. Acceptable values are 'precip' (the default), 'temp', 'rh', 'wind', 'qli' and 'qli'.  |
| quiet       | Optional. Suppresses display of messages, except for errors. Because this function can be very slow to execute, the default value is FALSE, to provide information on the downloading. |

## Value

Writes the specified files to the destination directory. If successful, returns TRUE. If unsuccessful, returns FALSE.

## Examples

```
## Not run:
NARRdownloadNetCDF('3h', 1979, 2015)
## End(Not run)
```



---

NARRgetNearestTimeseries

*Finds timeseries of NARR data nearest to specified location*


---

## Description

Reads a NARR NetCDF file, which contains all of the 3-hour values of a single variable for a single year, and extracts the values for a single location. Note that the values are *NOT* quality controlled - negative precipitation values are possible.

## Usage

```
NARRgetNearestTimeseries(ncdfFile, varName, pointLon, pointLat, timezone = "",
  quiet = TRUE, logfile = "")
```

## Arguments

|          |  |
|----------|--|
| ncdfFile | Name of the NetCDF file containing ERA data.   |
| varName  | Required. Name of the NetCDF variable to extract.  |
| pointLon | Required. Decimal longitude of desired location. Note that the NetCDF longitude is <i>East</i> , your value should be negative.  |
| pointLat | Required. Decimal latitude of desired location.  |
| timezone | Required. The name of the timezone of the data as a character string. This should be the timezone of your data, but omitting daylight savings time. Note that the timezone code is specific to your OS. To avoid problems, you should use a timezone without daylight savings time. Under Linux, you can use 'CST' and 'MST' for Central Standard or Mountain Standard time, respectively. Under Windows or OSX, you can use 'etc/GMT+6' or 'etc/GMT+7' for Central Standard and Mountain Standard time. DO NOT use 'America/Regina' as the time zone, as it includes historical changes between standard and daylight savings time. |
| quiet    | Optional. Suppresses display of messages, except for errors. If you are calling this function in an R script, you will usually leave quiet=TRUE (i.e. the default). If you are working interactively, you will probably want to set quiet=FALSE, as it will list the actual latitude and longitude of the ERA tile.  |
| logfile  | Optional. Name of the file to be used for logging the action. Normally not used.   |

## Value

If unsuccessful, returns FALSE. If successful, returns a standard **CRHMr** dataframe containing the datetime and the extracted data, which are unpacked (i.e. the NetCDF multiplier and offset have been applied). If quiet=FALSE, then the NARR location used and its distance to the specified location will be displayed.

If unsuccessful, returns FALSE. If successful, returns a **CRHMr** data frame of the specified variable. Note that no unit conversions are done, so the variable will be in the NARR units.

## Examples

```
## Not run:
p1979 <- NARRgetNearestTimeseries('acpcp.1979.nc', varName='acpcp', pointLon = -113,
pointLat = 52, timezone = 'CST', quiet=FALSE)
## End(Not run)
```

---

WATCHcreateHourlyWFDEIobs

*Creates a CRHM .obs file of hourly values from WATCH reanalysis data WFDEI files*

---

## Description

Extracts data from WATCH WFDEI netCDF files and builds a CRHM .obs file of hourly values containing t, ea, u10, p Qsi and Qli. All values are interpolated from 3 and 6 hur data. The windspeeds are at 10m, hence they are denoted as u10. Air temperatures are at 2m. The values for ea are computed from the atmospheric pressure (at 10m) and the absolute humidity (at 2m). Because the original NetCDF files are very large, this function typically runs very slowly. Also, because this function assembles and processes all of the data in memory, it can require a lot of RAM to execute.

## Usage

```
WATCHcreateHourlyWFDEIobs(nc.location = "", startyear = 1979,
endyear = 2012, lon = 0, lat = 0, precipType = "CRU",
sunTimeOffset = 2, solarMethod = "simpleMaxSolar",
interpolationMethod = "linear", obsFileName = "", timezone = "",
quiet = TRUE, logfile = "")
```

## Arguments

|               |  |
|---------------|--|
| nc.location   | Required. A character string of the directory holding the WATCH WFDEI netCDf files. This is a file path WITHOUT a terminal slash, e.g. 'z:\WATCH\WFDEI'  |
| startyear     | Optional. Year to begin. Must be in the range 1979–2012. Default is 1979.  |
| endyear       | Optional. Year to end. Must be in the range 1979–2012. Default is 2012.  |
| lon           | Required. Decimal longitude to extract for.  |
| lat           | Required. Decimal latitude to extract for.   |
| precipType    | Optional. The precipitation type used; can be 'CRU' (the default) or 'GPCC'.   |
| sunTimeOffset | Optional. Number of hours that local noon is offset from solar noon. The default is 2 hours. This may be incorrect. It seems that WATCH data may not incorporate the temporal offset correctly - zero may work better.   |
| solarMethod   | The method to be used for calculating the extra-terrestrial radiation. The default method is 'simpleMaxSolar'. Note that this method is only valid for latitudes between 49 and 55°N. The other supported method is 'PotSolarInst', which requires the package <b>EcoHydRology</b> to be installed |

|                     |   |
|---------------------|---|
| interpolationMethod | Optional. A vector containing the methods to be used for interpolation for each of the variables. Currently supported methods are 'linear' and 'spline'. The default is to use linear interpolation. If fewer methods than columns are specified, the methods are recycled.   |
| obsFileName         | Required. Name of the .obs file to be created.  |
| timezone            | Required. The name of the timezone of the data as a character string. This should be the timezone of your data, but omitting daylight savings time. Note that the timezone code is specific to your OS. To avoid problems, you should use a timezone without daylight savings time. You can use 'Etc/GMT+6' or 'Etc/GMT+7' for Central Standard and Mountain Standard time. DO NOT use 'America/Regina' as the time zone, as it includes historical changes between standard and daylight savings time. |
| quiet               | Optional. Suppresses display of messages, except for errors. If you are calling this function in an R script, you will usually leave quiet=TRUE (i.e. the default). If you are working interactively, you will probably want to set quiet=FALSE.  |
| logfile             | Optional. Name of the file to be used for logging the action. Normally not used.  |

### Value

If successful, returns the value TRUE and writes the specified .obs file. If unsuccessful, returns the value FALSE.

### Note

Because this function can be slow, and uses a lot of memory, you may wish to run it repeatedly for short intervals. You can then use the function appendObs in **CRHMr** to join the files together. The time shifting results in obs files which do not begin and end on day boundaries, so you should use the function trimObs in **CRHMr** to trim the obs file - *after* the final file has been assembled.

### Author(s)

Kevin Shook.

### References

R code for conversion of air pressure and absolute humidity was taken from project PEcAn The Predictive Ecosystem Analyzer <http://pecanproject.github.io>. The source code is available at <https://github.com/PecanProject/pecan/blob/master/modules/data.atmosphere/R/metutils.R>.

### See Also

[WATCHcreateHourlyWFDobs](#) [trimObs](#) [appendObs](#)

## Examples

```
## Not run:
location <- 'z:\data\WATCH\WFDEI'
obsName <- 'VermilionWATCH_WFDEI.obs'
WATCHcreateHourlyWFDobs(nc.location=location,
startyear=1979, endyear=2001, lon=-111.9, lat=53.2, timezone='Etc/GMT+7', obsFileName=obsName)
# read in file and trim it
obs <- CRHMr::readObsFile(obsFile=obsName, timezone='Etc/GMT+7')
trimmedObs <- CRHMr::trimObs(obs)
CRHMr::writeObsFile(trimmedObs, obsFileName)
## End(Not run)
```

---

WATCHcreateHourlyWFDobs

*Creates a CRHM .obs file of hourly values from WATCH reanalysis data WFD files*

---

## Description

Extracts data from WATCH WFD netCDF files and builds a CRHM .obs file of hourly values containing t, ea, u10, p Qsi and Qli. All values are interpolated from 3 and 6 hr data. The windspeeds are at 10m, hence they are denoted as u10. Air temperatures are at 2m. The values for ea are computed from the atmospheric pressure (at 10m) and the absolute humidity (at 2m). Because the original NetCDF files are very large, this function typically runs very slowly. Also, because this function assembles and processes all of the data in memory, it can require a lot of RAM to execute.

## Usage

```
WATCHcreateHourlyWFDobs(nc.location = "", startyear = 1901,
endyear = 2001, lon = 0, lat = 0, sunTimeOffset = 2,
solarMethod = "simpleMaxSolar", interpolationMethod = "linear",
obsFileName = "", timezone = "", quiet = TRUE, logfile = "")
```

## Arguments

|               |  |
|---------------|--|
| nc.location   | Required. A character string of the directory holding the WATCH WFD netCDF files. This is a file path WITHOUT a terminal slash, e.g. 'z:\WATCH\WFD'. |
| startyear     | Optional. Year to begin. Must be in the range 1901–2001. Default is 1901.  |
| endyear       | Optional. Year to end. Must be in the range 1901–2001. Default is 2001.  |
| lon           | Required. Decimal longitude to extract for.  |
| lat           | Required. Decimal latitude to extract for.   |
| sunTimeOffset | Optional. Number of hours that local noon is offset from solar noon. The default is 2 hours.   |

|                     |   |
|---------------------|---|
| solarMethod         | The method to be used for calculating the extra-terrestrial radiation. The default method is 'simpleMaxSolar'. Note that this method is only valid for latitudes between 49 and 55°N. The other supported method is 'PotSolarInst', which requires the package <b>EcoHydRology</b> to be installed  |
| interpolationMethod | Optional. A vector containing the methods to be used for interpolation for each of the variables. Currently supported methods are 'linear' and 'spline'. The default is to use linear interpolation. If fewer methods than columns are specified, the methods are recycled.   |
| obsFileName         | Required. Name of the .obs file to be created.  |
| timezone            | Required. The name of the timezone of the data as a character string. This should be the timezone of your data, but omitting daylight savings time. Note that the timezone code is specific to your OS. To avoid problems, you should use a timezone without daylight savings time. You can use 'Etc/GMT+6' or 'Etc/GMT+7' for Central Standard and Mountain Standard time. DO NOT use 'America/Regina' as the time zone, as it includes historical changes between standard and daylight savings |
| quiet               | Optional. Suppresses display of messages, except for errors. If you are calling this function in an R script, you will usually leave quiet=TRUE (i.e. the default). If you are working interactively, you will probably want to set quiet=FALSE.  |
| logfile             | Optional. Name of the file to be used for logging the action. Normally not used.  |

### Value

If successful, returns the value TRUE and writes the specified .obs file. Each month's data is written as it is created. If unsuccessful, returns the value FALSE.

### Author(s)

Kevin Shook

### References

R code for conversion of air pressure and absolute humidity was taken from project PEcAn The Predictive Ecosystem Analyzer <http://pecanproject.github.io>. The source code is available at <https://github.com/PecanProject/pecan/blob/master/modules/data.atmosphere/R/metutils.R>.

### Examples

```
## Not run:
location <- 'z:\data\WATCH\WFD'
obsFileName <- 'VermilionWATCH_WFD.obs'
WATCHcreateWFDobs(location, 1979, 2001, -111.9, 53.2, timezone='Etc/GMT+7', obsFileName=obsFileName)
## End(Not run)
```

---

|                     |   |
|---------------------|---|
| WATCHcreateWFDEIobs | <i>Creates a CRHM .obs file of 3-hourly values from WATCH reanalysis data WFDEI files</i> |
|---------------------|---|

---

### Description

Extracts data from WATCH WFDEI netCDF files and builds a CRHM .obs file of 3-hour data containing t, ea, u10, and p values. The output values can be interpolated to hourly using the function HourlyWATCHobs. The windspeeds are at 10m, hence they are denoted as u10. Air temperatures are at 2m. The values for ea are computed from the atmospheric pressure (at 10m) and the absolute humidity (at 2m).

### Usage

```
WATCHcreateWFDEIobs(nc.location = "", startyear = 1979, endyear = 2012,
  lon = 0, lat = 0, houroffset = 0, obsFileName = "", quiet = TRUE,
  logfile = "")
```

### Arguments

|             |  |
|-------------|--|
| nc.location | Required. A character string of the directory holding the WATCH WFDEI netCDF files. This is a file path WITHOUT a terminal slash, e.g. 'z:\WATCH\WFDEI'  |
| startyear   | Optional. Year to begin. Must be in the range 1979–2012. Default is 1979.  |
| endyear     | Optional. Year to end. Must be in the range 1979–2012. Default is 2012.  |
| lon         | Required. Decimal longitude to extract for.  |
| lat         | Required. Decimal latitude to extract for.   |
| houroffset  | Required. Number of hours that the local location is offset from UTC (GMT). Must be negative in the western hemisphere. For Mountain Standard Time, the offset is -7 hours.  |
| obsFileName | Required. Name of the .obs file to be created.   |
| quiet       | Optional. Suppresses display of messages, except for errors. If you are calling this function in an R script, you will usually leave quiet=TRUE (i.e. the default). If you are working interactively, you will probably want to set quiet=FALSE. |
| logfile     | Optional. Name of the file to be used for logging the action. Normally not used.   |

### Value

If successful, returns the value TRUE and writes the specified .obs file. Each month's data is written as it is created. If unsuccessful, returns the value FALSE.

### Author(s)

Kevin Shook.

## References

R code for conversion of air pressure and absolute humidity was taken from project PEcAn The Predictive Ecosystem Analyzer <http://pecanproject.github.io>. The source code is available at <https://github.com/PecanProject/pecan/blob/master/modules/data.atmosphere/R/metutils.R>.

## Examples

```
## Not run:
location <- 'z:\data\WATCH\WFDEI'
obsFileName <- 'VermilionWATCH_WFDEI.obs'
WATCHcreateWFDEIobs(location, 1979, 2001, -111.9, 53.2, -7, obsFileName)
## End(Not run)
```

---

|                   |  |
|-------------------|--|
| WATCHcreateWFDobs | <i>Creates a CRHM .obs file of 3-hourly and 6-hourly values from WATCH reanalysis data WFD files</i> |
|-------------------|--|

---

## Description

Extracts data from WATCH WFD netCDF files and builds a CRHM .obs file of 3-hour data containing t, ea, u10, and p values. The values of t, ea and u are 6-hourly, with NA values inserted. The data are output at MST. The output values can be interpolated to hourly values using the function HourlyWATCHObs. The windspeeds are at 10m, so they are denoted as u10. Air temperatures are at 2m. The values for ea are computed from the atmospheric pressure (at 10m) and the absolute humidity (at 2m).

## Usage

```
WATCHcreateWFDobs(nc.location = "", startyear = 1901, endyear = 2001,
  lon = 0, lat = 0, houroffset = 0, obsFileName = "", quiet = TRUE,
  logfile = "")
```

## Arguments

|             |   |
|-------------|---|
| nc.location | Required. A character string of the directory holding the WATCH WFD netCDF files. This is a file path WITHOUT a terminal slash, e.g. 'z:\WATCH\WFD'.                        |
| startyear   | Optional. Year to begin. Must be in the range 1901–2001. Default is 1901.   |
| endyear     | Optional. Year to end. Must be in the range 1901–2001. Default is 2001.   |
| lon         | Required. Decimal longitude to extract for.   |
| lat         | Required. Decimal latitude to extract for.  |
| houroffset  | Required. Number of hours that the local location is offset from UTC (GMT). Must be negative in the western hemisphere. For Mountain Standard Time, the offset is -7 hours. |
| obsFileName | Required. Name of the .obs file to be created.  |

|         |  |
|---------|--|
| quiet   | Optional. Suppresses display of messages, except for errors. If you are calling this function in an R script, you will usually leave quiet=TRUE (i.e. the default). If you are working interactively, you will probably want to set quiet=FALSE. |
| logfile | Optional. Name of the file to be used for logging the action. Normally not used.   |

**Value**

If successful, returns the value TRUE and writes the specified .obs file. Each month's data is written as it is created. If unsuccessful, returns the value FALSE.

**Author(s)**

Kevin Shook

**References**

R code for conversion of air pressure and absolute humidity was taken from project PEcAn The Predictive Ecosystem Analyzer <http://pecanproject.github.io>. The source code is available at <https://github.com/PecanProject/pecan/blob/master/modules/data.atmosphere/R/metutils.R>.

**Examples**

```
## Not run:
location <- 'z:\data\WATCH\WFD'
obsFileName <- 'VermilionWATCH_WFD.obs'
WATCHcreateWFDobs(location, 1979, 2001, -111.9, 53.2, -7, obsFileName)
## End(Not run)
```

---

WATCHdailyArealPrecip *Calculates daily WATCH areal precipitation*

---

**Description**

Calculates daily WATCH areal precipitation

**Usage**

```
WATCHdailyArealPrecip(monthlyPrecip, timezone)
```

**Arguments**

|               |  |
|---------------|--|
| monthlyPrecip | All WATCH precipitation for a month, as returned by WATCHgetWFDarealPrecip.  |
| timezone      | Required. The name of the timezone of the data as a character string. This should be the timezone of your data, but omitting daylight savings time. Note that the timezone code is specific to your OS. To avoid problems, you should use a timezone without daylight savings time. Under Linux, you can use 'CST' and 'MST' for Central Standard or Mountain Standard time, respectively. Under |



Windows or OSX, you can use 'etc/GMT+6' or 'etc/GMT+7' for Central Standard and Mountain Standard time. DO NOT use 'America/Regina' as the time zone, as it includes historical changes between standard and daylight savings time.

## Details

Calculates the daily total WATCH precipitation for all locations in a region.

## Value

If successful, returns a data frame consisting of the date, longitude, latitude, and total precipitation (in mm) for each point. If unsuccessful, returns FALSE.

## Examples

```
## Not run:
daily_rain <- WATCHdailyArealPrecip(monthly_rain, 'CST')
## End(Not run)
```

---

WATCHgetWFDarealPrecip

*Extracts the WATCH WFD precipitation for a specified domain*

---

## Description

Extracts the WATCH WFD precipitation (rainfall or snowfall) from a monthly netCDF file for a specified domain.

## Usage

```
WATCHgetWFDarealPrecip(ncdfFile = "", minLon = 0, maxLon = 0,
  minLat = 0, maxLat = 0, logfile = "")
```

## Arguments

|          |  |
|----------|--|
| ncdfFile | Required. NetCDF file containing monthly WATCH WFD precipitation (rainfall or snowfall). The file name must begin with the precipitation type (i.e. 'Rainf' or 'Snowf' and must end with the year and month, as in '190101'. |
| minLon   | Required. Minimum longitude for area to be extracted.  |
| maxLon   | Required. Maximum longitude for area to be extracted.  |
| minLat   | Required. Minimum latitude for area to be extracted.   |
| maxLat   | Required. Maximum latitude for area to be extracted.   |
| logfile  | Optional. Name of the file to be used for logging the action. Normally not used  |

**Value**

If successful, returns a data frame containing all of the precipitation values. The rows contain the values for each location for all time intervals. The first 2 columns contain the Longitude and Latitude, respectively for each location within the domain. The following columns contain the precipitation rate for each time interval. The names of these columns are the time intervals, e.g. '1901-01-01\_00:00' (note the underscore) in GMT.

**Author(s)**

Kevin Shook

**Examples**

```
## Not run:
monthly_rain <- WATCHgetWFDarealPrecip('Rainf_WFD_CRU_190101.nc',
  minLon=-105, maxLon=-95, minLat = 49, maxLat = 56)
## End(Not run)
```

---

WATCHgroupWFDEIobs      *Create obs files from WATCH WFDEI data for groups of sites*

---

**Description**

Extracts all variables from WFDEI NetCDF files and builds obs files for all specified locations

**Usage**

```
WATCHgroupWFDEIobs(siteFile, ncLocation, outputLocation, startyear = 1979,
  endyear = 2001, solarMethod = "PotSolarInst",
  interpolationMethod = "linear", quiet = TRUE, logfile = "")
```

**Arguments**

|            |  |
|------------|--|
| siteFile   | Required. A .csv file containing all of the variables required to describe the site locations. These are:<br><b>Name</b> Name of site<br><b>Land</b> Land ID number<br><b>Longitude</b> Site longitude<br><b>Latitude</b> Site latitude<br><b>glon</b> Site glon value<br><b>glat</b> Site glat value<br><b>timezone</b> Time zone for site. Must be in Etc format, e.g. Etc/GMT+7<br><b>SolarOffset</b> Local offset of solar noon in hours |
| ncLocation | Required. Location of all of the WATCH WFD files. Must have a trailing '\'   |

|                     |   |
|---------------------|---|
| outputLocation      | Required. Location for all of the output files. Must have a trailing '\`' symbol.   |
| startyear           | Optional. Year to begin extraction. Default is 1901.  |
| endyear             | Optional. Year to begin extraction. Default is 2001.  |
| solarMethod         | The method to be used for calculating the extra-terrestrial radiation. The default method is 'PotSolarInst', which requires the package <b>EcoHydRology</b> to be installed. The other supported method is 'simpleMaxSolar'. Note that this method is only valid for latitudes between 49 and 55°N. |
| interpolationMethod | Optional. A vector containing the methods to be used for interpolation for each of the variables. Currently supported methods are 'linear' and 'spline'. The default is to use linear interpolation. If fewer methods than columns are specified, the methods are recycled.                         |
| quiet               | Optional. Suppresses display of messages, except for errors. If you are calling this function in an R script, you will usually leave quiet=TRUE (i.e. the default). If you are working interactively, you will probably want to set quiet=FALSE.  |
| logfile             | Optional. Name of the file to be used for logging the action. Normally not used.  |

### Value

If successful, returns the value TRUE and writes the specified .obs file. Each month's data is written as it is created. If unsuccessful, returns the value FALSE.

### Author(s)

Kevin Shook

### References

R code for conversion of air pressure and absolute humidity was taken from project PEcAn The Predictive Ecosystem Analyzer <http://pecanproject.github.io>. The source code is available at <https://github.com/PecanProject/pecan/blob/master/modules/data.atmosphere/R/metutils.R>.

### Examples

```
## Not run:
points <- './Reanalysis/testdata/WATCH_selected_points.csv'
outputLocation <- './Reanalysis/testdata/'
ncloc <- '//water.usask.ca/Centre/Reanalysis_data/WATCH/WFD'
WATCHgroupWFDEIobs(siteFile = points, ncLocation = ncloc, outputLocation=outputLocation,
startyear = 1962, endyear = 1962))

## End(Not run)
```

WATCHgroupWFDobs

*Create obs files from WATCH WFD data for groups of sites***Description**

Extracts all variables from WFD NetCDF files and builds obs files for all specified locations

**Usage**

```
WATCHgroupWFDobs(siteFile, ncLocation, outputLocation, startyear = 1901,
  endyear = 2001, solarMethod = "PotSolarInst",
  interpolationMethod = "linear", quiet = TRUE, logfile = "")
```

**Arguments**

|                     |  |
|---------------------|--|
| siteFile            | Required. A .csv file containing all of the variables required to describe the site locations. These are:<br><b>Name</b> Name of site<br><b>Land</b> Land ID number<br><b>Longitude</b> Site longitude<br><b>Latitude</b> Site latitude<br><b>glon</b> Site glon value<br><b>glat</b> Site glat value<br><b>timezone</b> Time zone for site. Must be in Etc format, e.g. Etc/GMT+7<br><b>SolarOffset</b> Local offset of solar noon in hours |
| ncLocation          | Required. Location of all of the WATCH WFD files. Must have a trailing '\'   |
| outputLocation      | Required. Location for all of the output files. Must have a trailing '\'   |
| startyear           | Optional. Year to begin extraction. Default is 1901.   |
| endyear             | Optional. Year to begin extraction. Default is 2001.   |
| solarMethod         | The method to be used for calculating the extra-terrestrial radiation. The default method is 'PotSolarInst', which requires the package <b>EcoHydRology</b> to be installed. The other supported method is 'simpleMaxSolar'. Note that this method is only valid for latitudes between 49 and 55°N.  |
| interpolationMethod | Optional. A vector containing the methods to be used for interpolation for each of the variables. Currently supported methods are 'linear' and 'spline'. The default is to use linear interpolation. If fewer methods than columns are specified, the methods are recycled.  |
| quiet               | Optional. Suppresses display of messages, except for errors. If you are calling this function in an R script, you will usually leave quiet=TRUE (i.e. the default). If you are working interactively, you will probably want to set quiet=FALSE.   |
| logfile             | Optional. Name of the file to be used for logging the action. Normally not used.   |

**Value**

If successful, returns the value TRUE and writes the specified .obs file. Each month's data is written as it is created. If unsuccessful, returns the value FALSE.

**Author(s)**

Kevin Shook

**References**

R code for conversion of air pressure and absolute humidity was taken from project PEcAn The Predictive Ecosystem Analyzer <http://pecanproject.github.io>. The source code is available at <https://github.com/PecanProject/pecan/blob/master/modules/data.atmosphere/R/metutils.R>.

**Examples**

```
## Not run:
points <- './Reanalysis/testdata/WATCH_selected_points.csv'
outputLocation <- './Reanalysis/testdata/'
ncloc <- '//water.usask.ca/Centre/Reanalysis_data/WATCH/WFD'
WATCHgroupWFDobs(siteFile = points, ncLocation = ncloc, outputLocation=outputLocation,
startyear = 1962, endyear = 1962))

## End(Not run)
```

---

WATCHhourlyObs

---

*Interpolates 3 and 6 hour WATCH variables to hourly values*


---

**Description**

Interpolates t, u10, ea, qsi, and qli values (6-hour in WFD, 3-hour in WFDEI) to hourly. The total p values are evenly divided over the hourly intervals. Outputs hourly t, u10, ea, qsi, qli, and p.

**Usage**

```
WATCHhourlyObs(infile = "", outfile = "", logfile = "")
```

**Arguments**

|         |  |
|---------|--|
| infile  | Required. Name of file created by CreateWFDobs or CreateWFDobs.                |
| outfile | Required. Hourly obs file to be created.                                       |
| logfile | Optional. Name of the file to be used for logging the action. Normally not use |

**Value**

If successful, returns the value TRUE and writes the specified .obs file of hourly values. If unsuccessful, returns the value FALSE.

**Author(s)**

Kevin Shook

**Examples**

```
## Not run:  
WATCHhourlyObs('VermilionWATCH_WFD.obs', 'VermilionWFDhourly.obs')  
WATCHhourlyObs('VermilionWATCH_WFDEI.obs', 'VermilionWFDEIhourly.obs')  
## End(Not run)
```

# Index

## \*Topic **datasets**

land, [15](#)

appendObs, [19](#)

distributeQsi, [13](#)

ERAdailyArealPrecip, [2](#)

ERAdaccum, [3](#), [7](#), [8](#), [11](#)

ERAggetArealPrecip, [4](#)

ERAggetMultipleLocationTimeseries, [5](#)

ERAggetNearestTimeseries, [7](#), [7](#), [9](#), [11](#),  
[13–15](#)

ERAhourlyAirtemp, [9](#), [14](#)

ERAhourlyLongwave, [10](#), [13](#)

ERAhourlyPrecip, [11](#)

ERAhourlyShortwave, [11](#), [12](#)

ERAhourlyVP, [13](#)

ERAhourlyWindspeed, [14](#)

interpolate, [9](#), [14](#), [15](#)

land, [15](#)

NARRdownloadNetCDF, [16](#)

NARRgetNearestTimeseries, [17](#)

Reanalysis-package, [2](#)

trimObs, [19](#)

WATCHcreateHourlyWFDEIobs, [18](#)

WATCHcreateHourlyWFDobs, [15](#), [19](#), [20](#)

WATCHcreateWFDEIobs, [22](#)

WATCHcreateWFDobs, [23](#)

WATCHdailyArealPrecip, [24](#)

WATCHgetWFDarealPrecip, [25](#)

WATCHgroupWFDEIobs, [26](#)

WATCHgroupWFDobs, [28](#)

WATCHhourlyObs, [29](#)