

## **Wikiprint Book**

**Title: EcomsUdg/RPackage/Functions**

**Subject: TracMeteo - EcomsUdg/RPackage/Functions**

**Version: 44**

**Date: 11/30/2021 04:21:48 PM**

## Table of Contents

makeNcmIDataset	3
dataInventory	3
loadSystem4	4

## makeNcmlDataset

### Description

Generates a NcML file from a collection of netCDF files.

### Usage

```
makeNcmlDataset(source.dir, ncml.file)
```

### Arguments

- `source.dir`: character string indicating a valid path of the directory containing the files
- `ncml.file`: character string indicating the NcML file name (and path, default to working directory), including the extension `.ncml`.

The output is a NcML file named as `file.name` which will be stored in the `output.dir`.

### Details

- All files of the same dataset should be put together in the same directory, indicated by the `source.dir` argument.
- Currently the function works only with netCDF (`.nc`) file collections.
- A number of useful recommendations regarding dataset naming are provided [?here](#)

### Value

Creates a NcML file at the specified location

### Notes

A NcML file is a [?XML](#) representation of netCDF metadata. This is approximately the same information one gets when dumping the header of a netCDF file (e.g. by typing on the terminal the command `ncdump -h`). By means of NcML it is possible to create virtual datasets by modifying and aggregating other datasets, thus providing maximum flexibility and ease of access to data stored in collections of files containing data from different variables/time slices. The function `makeNcmlDataset` is intended to deal with reanalysis, forecasts and other climate data products, often consisting of collections of netCDF files corresponding to different variables and partitioned by years/decades or other time slices. It operates by applying to types of [?aggregation operations](#):

1. `Union`: Performs the union of all the dimensions, attributes, and variables in multiple NetCDF files
2. `JoinExisting`: Variables of the same name (in different files) are connected along their existing, outer dimension, called the aggregation dimension. In this case the aggregation dimension is `time`.

## dataInventory

### Description

Provides summary information about the main characteristics of a NcML dataset.

### Usage

```
dataInventory(ncml.file)
```

### Arguments

- `ncml.file`: a character string indicating the full path to the virtual dataset (the `NcML` file). This can be either a path containing the directory and name of the file, or an appropriate URL in case the dataset is remotely accessed (e.g., via the [?SPECS-EUPORIAS THREDDS](#)).

### Value

The output of the function consists of a list of variable length, depending on the number of variables contained in the dataset, following this structure:

- `Description`: Description of the variable
- `DataType`: Character string indicating data type (i.e. float ...)
- `Units`: Character string indicating the units of the variable

- **TimeStep:** A `difftime` class object representing the time interval between consecutive values in the time dimension axis
- Dimensions:** A list of length  $n$ , containing the following information for each of the  $n$  dimensions:
  - **Type:** Character vector indicating the type of dimension (e.g. Time, Lon, Pressure ...)
  - **Units:** Character vector indicating the units of the dimension axis
  - **Values:** A vector containing all the dimension values. This might be a vector of `POSIXlt` class in case of a dimension of type *time*, or numeric in other cases.

Details

A common need prior to data analysis is to get an overview of all data available and their structure (variables, dimensions, units, geographical extent, time span ...). Note that the function provides an overview of the raw data as they are stored in the original data files. The units may change after loading the function if conversions are applied via dictionary.

**loadSystem4**

Description

Loads hindcast/forecast data from ECMWF's System4 model by remotely accessing the SPECS-EUPORIAS THREDDS Data Server.

Usage

```
loadSystem4(dataset, var, members, lonLim, latLim, season, years, leadMonth)
```

Arguments

- **dataset:** A character string indicating the full URL path to the OPeNDAP dataset. Currently, the accepted values correspond to the available [?Datasets System4 datasets](#) `/System4_Seasonal_15Members.ncml`, but using the `System4_Seasonal_15Members.ncml`, `System4_Seasonal_51Members.ncml` or `System4_Annual_15Members.ncml` ending strings depending on the dataset of choice.
- **var:** Variable code. Argument values currently accepted are `tas`, `tasmin`, `tasmax`, `pr` or `mslp`, as internally defined in the vocabulary for System4 following the nomenclature displayed in the table below. However, note that new variables and datasets will be progressively included. Note that depending on the time step of the variable the units might be referred to different time aggregations. For instance, currently `mslp` is 6-hourly, and thus the 6-hourly mean value is returned for each time step. Similarly, 24-h accumulated values are returned for `pr`, and so on. Note that the *instantaneous* and *aggregated* fields in table below refer to the potential time step values that the variables may take, which does not mean that the resolution provided by the System4 model is necessarily that.

Short Name	Long name	Units	Instantaneous	Aggregated
tasmax	Maximum temperature at 2 metres	degC	No	Yes
tasmin	Minimum temperature at 2 metres	degC	No	Yes
tas	Mean temperature at 2 metres	degC	Yes	Yes
pr	Total precipitation accumulated	mm	No	Yes
mslp	Mean sea level pressure	Pa	Yes	Yes

- **members:** Optional. Default to all members. In the above case, a single member (the first) of the System4 ensemble is loaded, but additional members could be also specified (e.g. `members=NULL` for all members, or `members=1:5` for the first five members).
- **lonLim:** Vector of length = 2, with minimum and maximum longitude coordinates, in decimal degrees, of the bounding box selected.
- **latLim:** Vector of length = 2, with minimum and maximum latitude coordinates, in decimal degrees, of the bounding box selected.
- **season:** A vector of integers specifying the desired season (in months, January=1, etc.) of analysis. Options include a single month (as in the above example) or a standard season (e.g. `period = c(12,1,2)` for standard Boreal winter, DJF).
- **years:** Optional. Default to all available years. Vector of years to select. Note that in cases with year-crossing seasons (e.g. winter DJF, `season = c(12,1,2)`), for a particular year period `years = 1981:2000`, by convention the first season would be DJF 1980/81, if available (otherwise a warning message is given).

- `leadMonth`: Lead month forecast time corresponding to the first month of the specified season. Note that `leadMonth = 1` for `season = 1` (January) corresponds to the December initialization forecasts. In this way the effect of the lead time forecast in the analysis of a particular season can be analyzed by just changing this parameter.

The *SPECS-EUPORIAS Data Portal* can be remotely accessed from R via the [?loadSystem4.R](#) function. Note that this function is part of a more comprehensive R package currently under development. This function automatically cares about the proper location of the right indices for data sub-setting across the different variable dimensions, given a few simple arguments for subset definition. In addition, instead of retrieving a NetCDF file that needs to be opened and read, the requested data is directly loaded into the current R working session, according to a particular structure described below, prior to data analysis and/or representation.

A worked example describing a multi-model selection of a dataset is presented in the tutorial, which can be downloaded [?here](#), or in the section [Examples?](#). The output returned by the function consists of a list with the following elements providing the necessary information for data representation and analysis:

- `VarName`: Character string indicating the variable long name, as defined in the vocabulary (see Table above)
- `VarUnits`: Character string. Units of the variable, as returned in `MemberData`
- `TimeStep`: A `diffTime` class object. Indicates the time span of each forecast time
- `MemberData`: This is a list of length  $n$ , where  $n$  = number of members of the ensemble selected by the `member` argument. Each element of the dataset is a 2-D matrix of  $i$  rows  $\times$   $j$  columns, of  $i$  forecast times and  $j$  grid-points
- `LatLonCoords`: A 2-D matrix of  $j$  rows (where  $j$  = number of grid points selected) and two columns corresponding to the latitude and longitude coordinates respectively.
- `RunDates`: A `POSIXlt` time object corresponding to the initialization times selected. There is an initialization time associated to each forecast time.  
`ForecastDates`: A list with two `POSIXlt` time elements of length  $i$ , corresponding to the rows of each matrix in `MemberData`. The list contain tow elements:
  - `Start`: Starting times of the verification period of the variable
  - `End`: End time of the verification period of the variable