# Package: magpie4 (via r-universe)

October 16, 2024

**Type** Package

**Title** MAgPIE outputs R package for MAgPIE version 4.x

**Version** 2.13.8

**Date** 2024-10-16

**Description** Common output routines for extracting results from the MAgPIE framework (versions 4.x).

**URL** `https://github.com/pik-piam/magpie4`, `https://doi.org/10.5281/zenodo.1158582`

**BugReports** `https://github.com/pik-piam/magpie4/issues`

**Depends** R (>= 3.5.0), magclass (>= 2.40)

**Imports** dplyr, gdx2 (>= 0.1.6), gms, luscale, lusweave, madrat (>= 3.10.0), magpiesets, methods, mip, mstools, nonparaeff, reshape2, rlang, utils, stringr, stats

**Suggests** covr, luplot, ncdf4, rworldmap (>= 1.3.8), terra, testthat

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Repository** https://pik-piam.r-universe.dev

**RemoteUrl** https://github.com/pik-piam/magpie4

**RemoteRef** HEAD

**RemoteSha** 6287089fe1b319d35b10b80f694bbf14ff29d97e

## Contents

---

`magpie4-package` *MAgPIE outputs R library for MAgPIE version 4.x*

---

## Description

Common output routines for extracting results from the MAgPIE framework (versions 4.x).

## Author(s)

Maintainer: Jan Philipp Dietrich <dietrich@pik-potsdam.de>

## See Also

Useful links:

- `https://github.com/pik-piam/magpie4`
- doi:10.5281/zenodo.1158582
- Report bugs at `https://github.com/pik-piam/magpie4/issues`

---

`addGeometry` *addGeometry*

---

## Description

Enriches land use data on cluster resolution geometry information as required for conversion by magclass::as.SpatVector

## Usage

```
addGeometry(x, clustermap)
```

## Arguments

| | |
|---|---|
| `x` | Landuse data on cluster/cell resolution as a magclass object |
| `clustermap` | A dataframe mapping with columns cluster, cell, and country |

## Value

A magclass object enriched with geometry information

### Author(s)

Jan Philipp Dietrich, Pascal Sauer

### Examples

```
## Not run:
landUse <- magpie4::land("fulldata.gdx", level = "cell")
clustermap <- readRDS(Sys.glob("clustermap_*.rds"))
landUseEnriched <- magpie4::addGeometry(landUse, clustermap)
attr(landUseEnriched, "geometry")
attr(landUseEnriched, "crs")

## End(Not run)
```

---

## `agEmployment` *agEmployment*

### Description

returns employment in crop+livestock production from MAgPIE results

### Usage

```
agEmployment(
  gdx,
  type = "absolute",
  detail = TRUE,
  level = "reg",
  file = NULL,
  dir = "."
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `type` | "absolute" for total number of people employed, "share" for share out of working age population |
| `detail` | if TRUE, employment is disaggregated to crop products, livestock products and (if available) mitigation measures, if FALSE only aggregated employment is reported |
| `level` | spatial aggregation to report employment ("iso", "reg", "glo" or "regglo", if type is "absolute" also "grid") |
| `file` | a file name the output should be written to using write.magpie |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |

## Value

employment in agriculture as absolute value or as percentage of working age population

## Author(s)

Debbora Leip

## Examples

```
## Not run:
x <- agEmployment(gdx)

## End(Not run)
```

---

`AgGDP` *AgGDP*

## Description

Reads data to calculate the agricultural GDP

## Usage

```
AgGDP(gdx, file = NULL, level = "reg")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |

## Value

A MAgPIE object containing values related with overall value of production [million US$17]

## Author(s)

Edna Molina Bacca

## Examples

```
  ## Not run:
    x <- AgGDP(gdx)

## End(Not run)
```

---

## `AgriResearchIntensity` *AgriResearchIntensity*

---

### Description

calculates Agricultural Research Intensity (Investment in AgR&D/Total GDP) from a MAgPIE gdx file

### Usage

```
AgriResearchIntensity(gdx, file = NULL, level = "reg")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | aggregation level, reg, glo or regglo, cell or grid |

### Author(s)

David M Chen

### Examples

```
## Not run:
x <- AgriResearchIntensity(gdx)

## End(Not run)
```

---

## `anthropometrics` *anthropometrics*

---

### Description

Calculates anthropometic indicators from the food demand model

### Usage

```
anthropometrics(
  gdx,
  indicator = "bodyheight",
  age = "adults",
  sex = FALSE,
  bmi_groups = FALSE,
  level = "iso",
```

```
  dir = ".",
  spamfiledirectory = "",
  final = TRUE,
  file = NULL,
  calibrated = TRUE
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `indicator` | bodyheight, bodyweight, bodyweight_healthy, BMI(Body Mass Index) or PAL (physical activity level) |
| `age` | if TRUE, demand is scaled down to age-groups and sex using food requirements |
| `sex` | if FALSE, female and male are aggregated, if sex, results are divided into males and females |
| `bmi_groups` | if TRUE, data is provided by BMI group |
| `level` | Level of regional aggregation; "iso" ISO country codes, "reg" (regional), "glo" (global) |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |
| `spamfiledirectory` | deprecated. please use `dir` instead |
| `final` | final results or preliminary results (the latter are the ones magpie uses for optimization before last iteration with demand model) |
| `file` | a file name the output should be written to using write.magpie |
| `calibrated` | if TRUE, uses the calibrated intake estimates for bodyweight estimation |

### Details

Demand definitions are equivalent to FAO Food supply categories

### Value

bodyweight (kg), bodyheight (cm), BMI or PAL as magpie objects

### Author(s)

Benjamin Leon Bodirsky

### Examples

```
  ## Not run:
    x <- anthropometrics(gdx)

## End(Not run)
```

## BII *Biodiversity intactness index*

### Description

calculates the area weighted biodiversity intactness index (BII) out of a MAgPIE gdx file

### Usage

```
BII(
  gdx,
  file = NULL,
  level = "glo",
  mode = "auto",
  landClass = "sum",
  spatialWeight = NULL,
  adjusted = FALSE,
  bii_coeff = NULL,
  side_layers = NULL,
  dir = "."
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | level of regional aggregation; "cell" (magpie cluster level), "reg" (regional), "glo" (global), "regglo" (regional and global), "iso" (country level), "grid" (0.5 degree grid cell level). |
| `mode` | "auto" (default), "from_grid", "MAgPIE" or "postprocessing".<br>• "MAgPIE" reports the BV based on values from the MAgPIE biodiversity module.<br>• "postprocessing" calculates the BV based on land information from MAgPIE (for versions where biodiversity module was not available yet).<br>• "auto" uses "MAgPIE" if available and falls back to "postprocessing" otherwise.<br>• "from_grid" calculates BII values from BII output and returns aggregated values at the aggregation level specified. |
| `landClass` | "all" returns average BII values for all land classes of ov_bv, "sum" returns the weighted BII over all land classes of ov44_bv_weighted. |
| `spatialWeight` | Spatial weight for aggregating BII values. Only relevant if mode is "from_grid", adjusted is TRUE, or level is either "grid" or "iso". |
| `adjusted` | if "TRUE", function returns adjusted BII values (results have been adjusted for primary and secondary other land). |

| | |
|---|---|
| `bii_coeff` | file containing BII coefficients. Only needed for mode = "postprocessing". NULL tries to automatically detected the file. |
| `side_layers` | file containing LUH2 side layers. NULL tries to automatically detected the file. |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |

## Details

Calculates global, regional and cluster-level biodiversity intactness index (BII)

## Value

Biodiversity intactness index (unitless)

## Author(s)

Patrick v. Jeetze, Florian Humpenoeder, Felicitas Beier

## Examples

```
## Not run:
x <- BII(gdx)

## End(Not run)
```

---

# `bioplasticDemand` *bioplasticDemand*

## Description

returns demand for bioplastic or demand for substrate for bioplastic production

## Usage

```
bioplasticDemand(
  gdx,
  type = "bioplastic",
  detail = FALSE,
  level = "regglo",
  file = NULL
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `type` | "bioplastic" for bioplastic demand, "substrate" for biomass demand as substrate for bioplastic production |
| `detail` | only relevant for type = "substrate". If TRUE, substrate demand is disaggregated by crop type, if FALSE only the aggregated demand is reported. |
| `level` | spatial aggregation to report bioplastic/substrate demand (only "reg" or "regglo") |
| `file` | a file name the output should be written to using write.magpie |

## Author(s)

Debbora Leip

## Examples

```
## Not run:
x <- bioplasticDemand(gdx)

## End(Not run)
```

---

# `bodyweight` *bodyweight*

## Description

Calculates the prevalence of underweight, normalweight, overweight (excluding obesity) and obesity. For more detailed body mass classifications see functions population or anthropometrics.

## Usage

```
bodyweight(
  gdx,
  level = "reg",
  age = FALSE,
  sex = FALSE,
  share = FALSE,
  dir = ".",
  spamfiledirectory = "",
  population = NULL
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | Level of regional aggregation; "iso" ISO country codes, "reg" (regional), "glo" (global) |
| `age` | if TRUE, demand is scaled down to age-groups and sex using food requirements |
| `sex` | if FALSE, female and male are aggregated, if sex, results are divided into males and females |
| `share` | if TRUE, data is provided by BMI group |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |
| `spamfiledirectory` | deprecated. please use `dir` instead |
| `population` | population information from GDX. Can be provided to speed up calculation process. Will be read from GDX, if not provided. |

### Details

Demand definitions are equivalent to FAO Food supply categories

### Value

MAgPIE object with mio people or share of people in each weight category

### Author(s)

Benjamin Leon Bodirsky

### Examples

```
  ## Not run:
    x <- bodyweight(gdx)

## End(Not run)
```

---

## `carbonHWP` *carbonHWP*

---

### Description

reads carbon stored in harvested timber out of a MAgPIE gdx file

## Usage

```
carbonHWP(
  gdx,
  file = NULL,
  level = "cell",
  unit = "element",
  half_life = 35,
  cumulative = FALSE,
  baseyear = 1995
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "cell", "reg" (regional), "glo" (global), "regglo" (regional and global) or any secdforest aggregation level defined in superAggregate |
| `unit` | element" or "gas"; "element": co2_c in Mt C/yr, n2o_n in Mt N/yr, ch4 in Mt CH4/yr; "gas": co2_c Mt CO2/yr, n2o_n in Mt NO2/yr, ch4 in Mt CH4/yr |
| `half_life` | Half life in years for decay in wood products loosing half theor carbon content. (35 yrs is deafault) |
| `cumulative` | Logical; Determines if cHWP emissions are reported annually (FALSE) or cumulative (TRUE). The starting point for cumulative emissions is y1995. |
| `baseyear` | Baseyear used for cumulative emissions (default = 1995) |

## Details

Annual (and cumulative) Carbon stored in harvested wood products as well as slow emissions from half life deacy.

## Value

carbon stocks in MtC from harvested timber

## Author(s)

Abhijeet Mishra, Florian Humpenoeder

## Examples

```
  ## Not run:
    x <- carbonHWP(gdx)

## End(Not run)
```

---

`carbonLTS` *carbonLTS*

---

## Description

reads carbon stored in harvested timber out of a MAgPIE gdx file

## Usage

```
carbonLTS(
  gdx,
  file = NULL,
  level = "cell",
  unit = "element",
  cumulative = FALSE,
  baseyear = 1995
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "cell", "reg" (regional), "glo" (global), "regglo" (regional and global) or any secdforest aggregation level defined in superAggregate |
| `unit` | element" or "gas"; "element": co2_c in Mt C/yr, n2o_n in Mt N/yr, ch4 in Mt CH4/yr; "gas": co2_c Mt CO2/yr, n2o_n in Mt NO2/yr, ch4 in Mt CH4/yr |
| `cumulative` | Logical; Determines if cHWP emissions are reported annually (FALSE) or cumulative (TRUE). The starting point for cumulative emissions is y1995. |
| `baseyear` | Baseyear used for cumulative emissions (default = 1995) |

## Details

Annual (and cumulative) Carbon stored in harvested wood products as well as slow emissions from half life deacy.

## Value

carbon stocks in MtC from harvested timber

## Author(s)

Abhijeet Mishra, Florian Humpenoeder

## Examples

```
## Not run:
x <- carbonLTS(gdx)

## End(Not run)
```

---

## `carbonstock` *carbonstock*

### Description

reads carbon stocks out of a MAgPIE gdx file

### Usage

```
carbonstock(
  gdx,
  file = NULL,
  level = "cell",
  sum_cpool = TRUE,
  sum_land = TRUE,
  subcategories = NULL,
  stockType = "actual"
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "cell", "reg" (regional), "glo" (global), "regglo" (regional and global) or any secdforest aggregation level defined in superAggregate |
| `sum_cpool` | sum over carbon pool dimension (default = TRUE) |
| `sum_land` | sum over land type dimension (default = TRUE) |
| `subcategories` | NULL or vector of strings. If NULL, no subcategories are returned. Meaningful options are "crop, "forestry" and "other" |
| `stockType` | carbon stock type (default = "actual"). Options: "actual", "previousLandPattern" and "previousCarbonDensity". |

### Details

carbon pools consist of vegetation carbon (vegc), litter carbon (litc) and soil carbon (soilc)

### Value

carbon stocks in MtC

### Author(s)

Florian Humpenoeder

### Examples

```
  ## Not run:
    x <- carbonstock(gdx)

## End(Not run)
```

---

## `cellularFit` *cellular fit*

### Description

cellular fit/error/bias calculations at regional and global level

### Usage

```
cellularFit(
  gdx,
  file = NULL,
  level = "cell",
  statistic = "MAE",
  variable = "land",
  dataset = "LUH2",
  water_aggr = FALSE
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | level at which the regional and global bias should be calculated. Options "cell" or "grid" |
| `statistic` | R2, MAE, MPE (mean percentage error - bias), MAPE (mean absolute percentage error) |
| `variable` | variable to be evaulated: land (land types) or crop (crop types) |
| `dataset` | dataset to compare with. LUH2 only option for variable land. LUH2 and MAPSPAM for the crop variable. |
| `water_aggr` | if irrigation types for crops should be agregated or not |

### Value

returns selected statistic at regglo level for the historical part of the time horizon

## Author(s)

Edna J. Molina Bacca, Patrick v. Jeetze

## Examples

```
  ## Not run:
    x <- cellularFit(gdx)

## End(Not run)
```

---

`checkLibrary` *modelstat*

## Description

Function to check if the library functions work with the newest magpie version

## Usage

```
checkLibrary(gdx, level = NULL)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |

## Details

This function simply tries to run all functions in the magpie library on the provided gdx file.

## Value

A list with three entries:

- `notest` Testing these functions was impossible because there wer missing arguments.
- `error` These functions could not be executed properly.
- `fine` Everything was fine with these functions.

## Author(s)

Markus Bonsch

### Examples

```
    ## Not run:
      x <- modelstat(gdx)

  ## End(Not run)
```

---

## clusterOutputToTerraVector

*Convert cluster output to terra vector*

### Description

Enriches land use data on cluster resolution with explicit spatial information by creating a terra polygon for each cluster according to the given clustermap.

### Usage

```
clusterOutputToTerraVector(x, clustermap)
```

### Arguments

| | |
|---|---|
| `x` | Landuse data on cluster/cell resolution as a magclass object |
| `clustermap` | A dataframe mapping with columns cluster, cell, and country |

### Value

A SpatVector with the following columns: c("clusterId", "country", "region", "year", "landtype", "value")

### Author(s)

Pascal Führlich, Patrick v. Jeetze

### Examples

```
## Not run:
landUse <- magpie4::land("fulldata.gdx", level = "cell")
clustermap <- readRDS(Sys.glob("clustermap_*.rds"))
clusterPolygons <- magpie4::clusterOutputToTerraVector(landUse, clustermap)
terra::writeVector(clusterPolygons, "cluster_resolution.shp")

## End(Not run)
```

---

## `consumptionValue` *consumptionValue*

---

### Description

calculates consumption value of different types based on a MAgPIE gdx file.

### Usage

```
consumptionValue(
  gdx,
  file = NULL,
  level = "reg",
  products = "kall",
  product_aggr = TRUE,
  type = NULL,
  type_aggr = TRUE
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `products` | Selection of products (either by naming products, e.g. "tece", or naming a set,e.g."kcr") |
| `product_aggr` | aggregate over products or not (boolean, default TRUE) |
| `type` | Consumption type(s): "food", "feed", "processed", "other_util", "bioenergy", "seed", "waste", "dom_balanceflow; NULL returns all types |
| `type_aggr` | aggregate over demand types or not (boolean, default TRUE) |

### Value

A MAgPIE object containing consumption value in million $US.

### Author(s)

Miodrag Stevanovic

### Examples

```
  ## Not run:
    x <- consumptionValue(gdx)

## End(Not run)
```

---

## `CostCapital` *CostCapital*

---

### Description

Reads data to calculate capital stocks

### Usage

```
CostCapital(gdx, type = "stocks", file = NULL, level = "cell")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `type` | either capital stocks ("stocks") or overall capital investment "investment" |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |

### Value

A MAgPIE object containing values related with overall value of production [million US$17]

### Author(s)

Edna Molina Bacca

### Examples

```
  ## Not run:
    x <- CostCapital(gdx)

## End(Not run)
```

---

## `costInputFactorsCrop` *costInputFactorsCrop*

---

### Description

Reads data to calculate Input factors with different approaches

### Usage

```
costInputFactorsCrop(gdx, type = "annuity", file = NULL, level = "reg")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `type` | Type of capital investments accounting. It can either be total investments ("investment"), or considering the annuity ("annuity") of the current time step. NULL in case the runs were not done with the sticky realization. |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation |

### Value

A MAgPIE object containing values related with overall value of production [million US$17]

### Author(s)

Edna Molina Bacca

### Examples

```
## Not run:
x <- costInputFactorsCrop(gdx)

## End(Not run)
```

---

## `CostOverall` *CostOverall*

### Description

Gross value of productions

### Usage

```
CostOverall(gdx, file = NULL, level = "reg")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |

### Value

A MAgPIE object containing values related with overall value of production [million US$17]

### Author(s)

Edna Molina Bacca

### Examples

```
## Not run:
x <- CostOverall(gdx)

## End(Not run)
```

---

## `costs` *costs*

---

### Description

reads costs entering the objective function from a MAgPIE gdx file

### Usage

```
costs(gdx, file = NULL, level = "reg", type = "annuity", sum = TRUE)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `type` | either "annuity" (as it enters the objetive function) or "investment" (investment) |
| `sum` | total costs (TRUE) or detailed costs (FALSE) |

### Value

A MAgPIE object containing the goal function costs including investments [million US$17]

### Author(s)

Jan Philipp Dietrich, Markus Bonsch, Misko Stevanovic, Florian Humpenoeder, Edna J. Molina Bacca, Michael Crawford

### Examples

```
## Not run:
x <- costs(gdx)

## End(Not run)
```

---

## `CostsAEI` *CostsAEI*

---

### Description

reads AEI costs entering the objective function from a MAgPIE gdx file

### Usage

```
CostsAEI(gdx, file = NULL, level = "regglo")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation ("reg", "glo", "regglo") |

### Value

MAgPIE object containing costs for AEI [million US$17]

### Author(s)

Felicitas Beier

### Examples

```
  ## Not run:
    x <- CostsAEI(gdx)

## End(Not run)
```

---

## `CostsFertilizer` *CostsFertilizer*

---

### Description

reads costs entering the objective function from a MAgPIE gdx file

### Usage

```
CostsFertilizer(gdx, file = NULL, level = "regglo", disagg = TRUE)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation ("reg", "glo", "regglo") |
| `disagg` | whether costs should be disaggregated into the different crop types |

## Value

MAgPIE object containing fertilizer costs [million US$17]

## Author(s)

Debbora Leip

## Examples

```
  ## Not run:
    x <- CostsFertilizer(gdx)

## End(Not run)
```

---

# `costsMACCS` *costsMACCS*

## Description

reads costs entering the objective function from a MAgPIE gdx file

## Usage

```
costsMACCS(gdx, file = NULL, level = "regglo")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation ("reg", "glo", "regglo") |

## Value

MAgPIE object containing mitigation costs [million US$17]

## Author(s)

Debbora Leip

## Examples

```
  ## Not run:
    x <- costsMACCS(gdx)

## End(Not run)
```

---

# costsPresolve *costsPresolve*

## Description

reads presovle costs (i.e. without bioenergy demand) entering the objective function from a MAgPIE gdx file

## Usage

```
costsPresolve(gdx, file = NULL, level = "reg")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |

## Details

Presolve is without bioenergy demand. Hence costs from a MAgPIE run with bioenergy demand minus costs from presolve reflect costs that can be attributed to bioenergy production

## Value

A MAgPIE object containing the goal function costs in presolve mode [million US$17]

## Author(s)

Florian Humpenoeder

## Examples

```
  ## Not run:
    x <- costsPresolve(gdx)

## End(Not run)
```

## `costsProductionCrops` *costsProductionCrops*

### Description

Reads data to calculate production costs for crops, costs related with investments are reported as annual average for both types (annuity, investment).

### Usage

```
costsProductionCrops(
  gdx,
  file = NULL,
  level = "regglo",
  type = "investment",
  dir = "."
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in gdxAggregate |
| `type` | Type of reporting, either "annuity" or total "investments" |
| `dir` | directory with mapping for disaggregation to higher resolutions |

### Value

A MAgPIE object containing values related with costs for crops production per ton produced [million US$17/tDM]

### Author(s)

Edna Molina Bacca

### Examples

```
## Not run:
x <- costsProductionCrops(gdx)

## End(Not run)
```

---

## `costsWholesale` *costsWholesale*

---

### Description

Reads data to calculate wholesale costs

### Usage

```
costsWholesale(gdx, file = NULL, level = "regglo")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |

### Value

A MAgPIE object containing values related with costs wholesale trade [million US$17/tDM]

### Author(s)

David M Chen

### Examples

```
## Not run:
x <- costsWholesale(gdx)

## End(Not run)
```

---

## `CostsWithoutIncentives` *CostsWithoutIncentives*

---

### Description

calculates agricultural costs without taxes, incentives and technical penalty costs (i.e. GHG taxes and BII incentives)

### Usage

```
CostsWithoutIncentives(gdx, file = NULL, level = "regglo")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | aggregation level, reg, glo or regglo |

### Value

A MAgPIE object containing the costs without taxes, incentives and technical penalty costs [million US$17]

### Author(s)

David M Chen

### Examples

```
## Not run:
x <- CostsWithoutIncentives(gdx)

## End(Not run)
```

---

## `CostTC` *CostTC*

### Description

Reads data on TC costs

### Usage

```
CostTC(gdx, file = NULL, level = "reg")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |

### Value

A MAgPIE object containing values related with overall value of production [million US$17]

### Author(s)

David Chen

### Examples

```
  ## Not run:
    x <- CostTC(gdx)

## End(Not run)
```

---

## CostTransport *CostTransport*

---

### Description

reads costs entering the objective function from a MAgPIE gdx file

### Usage

```
CostTransport(gdx, file = NULL, level = "cell", sum = FALSE)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `sum` | total costs (TRUE) or detailed costs (FALSE) |

### Value

A MAgPIE object containing the transport costs [million US$17]

### Author(s)

David Chen

### Examples

```
  ## Not run:
    x <- CostTransport(gdx)

## End(Not run)
```

---

`croparea` *croparea*

---

## Description

reads croparea out of a MAgPIE gdx file. Croparea excludes fallow land.

## Usage

```
croparea(
  gdx,
  file = NULL,
  level = "reg",
  products = "kcr",
  product_aggr = TRUE,
  water_aggr = TRUE,
  dir = ".",
  spamfiledirectory = ""
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in gdxAggregate |
| `products` | Selection of products (either by naming products, e.g. "tece", or naming a set,e.g."kcr") |
| `product_aggr` | aggregate over products or not (boolean) |
| `water_aggr` | aggregate irrigated and non-irriagted production or not (boolean). |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |
| `spamfiledirectory` | deprecated. please use `dir` instead |

## Value

production as MAgPIE object (unit depends on attributes)

## Author(s)

Jan Philipp Dietrich, Florian Humpenoeder

## See Also

`reportCroparea`

## Examples

```
## Not run:
x <- croparea(gdx)

## End(Not run)
```

---

`CropareaDiversityIndex`

# *CropareaDiversityIndex*

---

## Description

calculates an index that measures the croparea diversity

## Usage

```
CropareaDiversityIndex(
  gdx,
  index = "shannon",
  level = "reg",
  measurelevel = "cell",
  groupdiv = "agg1",
  dir = "."
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `index` | can be "shannon", "gini" or "invsimpson" for different types of diversitiy indices |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `measurelevel` | level at which diversity is measured. "cell" means diversity |
| `groupdiv` | should crop groups be split up into several individual items or not? Choose either FALSE or different (dis)aggregation methods "agg1", "agg2" |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation is measured at cellular level, even if lateron average diversity is aggregated to regional level. |

## Value

MAgPIE object (unit depends on attributes)

## Author(s)

Benjamin Leon Bodirsky

## See Also

`CropareaDiversityIndex`

## Examples

```
## Not run:
x <- CropareaDiversityIndex(gdx)

## End(Not run)
```

---

# `croplandTreeCover` *croplandTreeCover*

## Description

calculates tree cover on cropland (Mha) from a MAgPIE gdx file

## Usage

```
croplandTreeCover(gdx, level = "reg", dir = ".", sum_ac = TRUE, debug = FALSE)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | aggregation level, reg, glo or regglo, cell or grid |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) |
| `sum_ac` | sum over age classes TRUE / FALSE |
| `debug` | debug mode TRUE makes some consistency checks between estimates for different resolutions |

## Author(s)

Florian Humpenoeder

## Examples

```
## Not run:
x <- fallow(gdx)

## End(Not run)
```

---

`cshare` *cshare*

---

## Description

Calculates soil carbon share in relation to potential natural vegetation based on a MAgPIE gdx file

## Usage

```
cshare(
  gdx,
  file = NULL,
  level = "reg",
  reference = "actual",
  noncrop_aggr = TRUE,
  dir = ".",
  spamfiledirectory = ""
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `reference` | default is "actual" (cshare in actual carbon stocks). Other option is "target" (cshare in target carbon stocks). |
| `noncrop_aggr` | aggregate non cropland types to 'noncropland' (if FALSE all land types of pools59 will be reported) |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |
| `spamfiledirectory` | deprecated. please use `dir` instead |

## Value

A MAgPIE object containing som values

## Author(s)

Kristine Karstens

## Examples

```
## Not run:
x <- cshare(gdx)

## End(Not run)
```

---

# deco *deco*

## Description

Function that quantifies the influences of the underlying drivers to a dependent output variable. It attributes the changes of the output variable (A) to changes of several drivers (B, B/C, C/A). The output must be the product of the drivers.

## Usage

```
deco(data, names_factor = NULL, plot = FALSE)
```

## Arguments

| | |
|---|---|
| `data` | Decomposition Data as a magpie object. The first column of the third dimension has to be the output (A), while the subsequent columns are the coefficients of the drivers (B,C,...). Example: Area = Population x Supply/Population x Area/Supply. 3rd-dimension column order then has to be: Area, Population, Supply. |
| `names_factor` | Names of the output (A) and the Decomposition-Factors (B,B/C,C/A), if names_factor=NULL the names for the third column will be generated like the factors for decomposition (above example: Area, Population, Supply/Population, Area/Supply) |
| `plot` | TRUE or FALSE |

## Details

Use function deco_plot in library luplot to make a plot out of this. It is only usable for the decomposition of 5 or less drivers. For documentation, see paper Huber, Veronika, Ina Neher, Benjamin L. Bodirsky, Kathrin Hoefner, and Hans Joachim Schellnhuber. 2014. "Will the World Run out of Land? A Kaya-Type Decomposition to Study Past Trends of Cropland Expansion." Environmental Research Letters 9 (2): 024011. https://doi.org/10.1088/1748-9326/9/2/024011. Or see master Thesis of Ina Neher (2013)

## Value

Decomposes the impact of certain drivers to an output (A) value.

## Author(s)

Ina Neher, Benjamin Leon Bodirsky

### Examples

```
Data<-array(c(1,1.1,1.15,1,1.05,1.1,1,1.05,1.15),c(3,3))
dimnames(Data)<-list(paste("y",2000:2002,sep=""),c("Area","Population","Supply"))
Data <- as.magpie(Data)
deco(Data)
```

---

## `demand` *demand*

### Description

Calculates MAgPIE demand out of a gdx file

### Usage

```
demand(
  gdx,
  file = NULL,
  level = "reg",
  products = "kall",
  product_aggr = FALSE,
  attributes = "dm",
  type = NULL,
  type_aggr = FALSE
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation ("reg", "glo", "regglo") |
| `products` | Selection of products (either by naming products, e.g. "tece", or naming a set,e.g."kcr") |
| `product_aggr` | aggregate over products or not (boolean, default FALSE) |
| `attributes` | dry matter: Mt ("dm"), gross energy: PJ ("ge"), reactive nitrogen: Mt ("nr"), phosphor: Mt ("p"), potash: Mt ("k"), wet matter: Mt ("wm"). Can also be a vector. |
| `type` | Demand type(s): "food", "feed", "processed", "other_util", "bioenergy", "seed", "waste", "dom_balanceflow; NULL returns all types |
| `type_aggr` | aggregate over demand types or not (boolean, default FALSE) |

### Details

Demand definitions are equivalent to FAO CBS categories

### Value

demand as MAgPIE object (Unit depends on attributes)

### Author(s)

Benjamin Leon Bodirsky, Abhijeet Mishra, Miodrag Stevanovic

### Examples

```
  ## Not run:
    x <- demand(level="regglo", products="kcr")

## End(Not run)
```

---

## `demandBioenergy` *demandBioenergy*

### Description

reads bioenergy demand from a MAgPIE gdx file

### Usage

```
demandBioenergy(gdx, file = NULL, level = "reg", sum = FALSE, round = NULL)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `sum` | 1st and 2nd generation bioenergy demand (FALSE) or total bioenergy demand (TRUE) |
| `round` | NULL or number of digits |

### Value

A MAgPIE object containing bioenergy demand in EJ/yr

### Author(s)

Jan Philipp Dietrich, Florian Humpenoeder

## Examples

```
  ## Not run:
    x <- demandBioenergy(gdx)

## End(Not run)
```

---

# `discountRates` *discountRates*

## Description

reads discount rates from a MAgPIE gdx file

## Usage

```
discountRates(gdx, file = NULL, level = "reg")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |

## Value

A MAgPIE object containing discount rates used in the model

## Author(s)

Xiaoxi Wang

## Examples

```
  ## Not run:
    x <- discountRates(gdx)

## End(Not run)
```

---

## `emisCO2` *emisCO2*

---

### Description

reads detailed CO2 emissions out of a MAgPIE gdx file

### Usage

```
emisCO2(
  gdx,
  file = NULL,
  level = "cell",
  unit = "gas",
  sum_cpool = TRUE,
  sum_land = TRUE,
  cumulative = FALSE,
  baseyear = 1995,
  lowpass = 3
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `unit` | "element" or "gas"; "element": co2_c in Mt C/yr "gas": co2_c Mt CO2/yr |
| `sum_cpool` | aggregate carbon pools (TRUE), below ground (soilc) and above ground (vegc and litc) will be reported, if FALSE |
| `sum_land` | TRUE (default) or FALSE. Sum over land types (TRUE) or report land-type specific emissions (FALSE). |
| `cumulative` | Logical; Determines if emissions are reported annually (FALSE) or cumulative (TRUE). The starting point for cumulative emissions is y1995. |
| `baseyear` | Baseyear used for cumulative emissions (default = 1995) |
| `lowpass` | number of lowpass filter iterations (default = 3) |

### Value

CO2 emissions as MAgPIE object (unit depends on `unit`)

### Author(s)

Florian Humpenoeder, Michael Crawford

## Examples

```
## Not run:
x <- emisCO2(gdx)

## End(Not run)
```

---

`Emissions` *Emissions*

---

## Description

reads GHG emissions out of a MAgPIE gdx file

## Usage

```
Emissions(
  gdx,
  file = NULL,
  level = "reg",
  type = "co2_c",
  unit = "element",
  subcategories = TRUE,
  cumulative = FALSE,
  lowpass = NULL,
  inorg_fert_split = TRUE
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `type` | emission type(s): "co2_c", "n2o_n" or "ch4" |
| `unit` | "element", "gas", "GWP100AR5", "GWP100AR6", "GWP*AR5", or "GWP*AR6" "element": co2_c in Mt C/yr, n2o_n in Mt N/yr, ch4 in Mt CH4/yr "gas": co2_c in Mt CO2/yr, n2o_n in Mt NO2/yr, ch4 in Mt CH4/yr "GWP": co2_c in Mt CO2/yr, n2o_n in Mt CO2eq/yr, ch4 in Mt CO2eq/yr |
| `subcategories` | FALSE (default) or TRUE |
| `cumulative` | Logical; Determines if emissions are reported annually (FALSE) or cumulative (TRUE). The starting point for cumulative emissions is y1995. |
| `lowpass` | number of lowpass filter iterations |
| `inorg_fert_split` | if TRUE then inorganic fertilizer emissions are further disaggregated into pasture- and cropland-related emissions. Both the aggregated ("inorg_fert") and disaggregated values ("inorg_fert_crop", "inorg_fert_past)" are reported |

### Value

emissions as MAgPIE object (unit depends on `unit`)

### Author(s)

Florian Humpenoeder, Benjamin Leon Bodirsky

### Examples

```
  ## Not run:
    x <- Emissions(gdx)

## End(Not run)
```

---

## `EmissionsBeforeTechnicalMitigation` *EmissionsBeforeTechnicalMitigation*

### Description

reads GHG emissions before technical abatement out of a MAgPIE gdx file. Technical abatement includes all abatement done in the MACC curves, but exclude endogenous mitigation. These emissions are NOT the standard reporting emissions, but used for special purposes like remind-magpie coupling.

### Usage

```
EmissionsBeforeTechnicalMitigation(
  gdx,
  file = NULL,
  level = "reg",
  type = "co2_c",
  unit = "element",
  subcategories = FALSE
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `type` | emission type(s): "co2_c", "n2o_n" or "ch4" and in the case of unit="gas" "co2" and "n2o" |

| | |
|---|---|
| `unit` | "element", "gas" or "co2eq"; "element": co2_c in Mt C/yr, n2o_n in Mt N/yr, ch4 in Mt CH4/yr; "gas": co2_c Mt CO2/yr, n2o_n in Mt NO2/yr, ch4 in Mt CH4/yr; "co2eq": co2_c in Mt CO2/yr, n2o_n in Mt CO2eq/yr, ch4 in Mt CO2eq/yr |
| `subcategories` | FALSE (default) or TRUE |

### Value

emissions as MAgPIE object (unit depends on `unit`)

### Author(s)

Florian Humpenoeder; Benjamin Leon Bodirsky

### Examples

```
  ## Not run:
    x <- EmissionsBeforeTechnicalMitigation(gdx)

## End(Not run)
```

---

## `expenditureIndexFood` *expenditureIndexFood*

---

### Description

calculates food expenditure index (baseyear = 100) corrected for ghg emission costs based on a MAgPIE gdx file

### Usage

```
expenditureIndexFood(
  gdx,
  file = NULL,
  level = "reg",
  products = "kfo",
  basketyear = "y2010",
  baseyear = "y2010",
  round = TRUE,
  ghgtax = TRUE
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | File the output should be written to using write.magpie |

| | |
|---|---|
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in mapping |
| `products` | Selection of products (either by naming products, e.g. "tece", or naming a set, e.g."kcr") |
| `basketyear` | Year of reference food basket (should be in the past for comparison of different runs to have identical and comparable food basket) |
| `baseyear` | Baseyear of the price index |
| `round` | Rounded result (TRUE or FALSE) |
| `ghgtax` | Correction of food price expenditure for ghg emission costs (TRUE or FALSE) |

### Value

A MAgPIE object containing food price expenditure index

### Author(s)

Felicitas Beier

### Examples

```
## Not run:
x <- expenditureIndexFood(gdx)

## End(Not run)
```

---

## `factorCosts` *factorCosts*

### Description

reads factor costs for crops, livestock, residues or pasture entering the objective function from a MAgPIE gdx file. Depending on the product and the MAgPIE version (and factor cost realization), factor costs are either already split into labor and capital, will be split in this function, or are kept as the aggregate

### Usage

```
factorCosts(gdx, products = "kli", file = NULL, level = "regglo")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `products` | products for which factor costs should be reported ("kcr", "kli", "kres", "fish", or "pasture") |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation ("reg", "glo", "regglo") |

### Value

MAgPIE object containing factor costs [million US$17]

### Author(s)

Debbora Leip

### Examples

```
## Not run:
x <- factorCosts(gdx)

## End(Not run)
```

---

## `factorCostShares` *factorCostShares*

### Description

returns labor and capital cost share out of factor costs (i.e. labor + capital)

### Usage

```
factorCostShares(
  gdx,
  type = "optimization",
  products = "kcr",
  level = "reg",
  file = NULL
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `type` | • "requirements": shares from factor requirements<br>• "optimization": cost shares between labor and capital costs in optimization<br>• "accounting": cost shares based on accounting of labor and capital costs |
| `products` | products for which cost shares should be reported, kcr or kli |
| `level` | spatial aggregation to report employment ("reg", "glo" or "regglo") |
| `file` | a file name the output should be written to using write.magpie |

### Value

labor and capital cost share out of factor costs

## Author(s)

Debbora Leip

## Examples

```
## Not run:
x <- factorCostShares(gdx)

## End(Not run)
```

---

`fallow` *fallow*

## Description

calculates fallow land (Mha) from a MAgPIE gdx file

## Usage

```
fallow(gdx, level = "reg", dir = ".", debug = FALSE)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | aggregation level, reg, glo or regglo, cell or grid |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) |
| `debug` | debug mode TRUE makes some consistency checks between estimates for different resolutions |

## Author(s)

Benjamin Leon Bodirsky

## Examples

```
## Not run:
x <- fallow(gdx)

## End(Not run)
```

---

`feed` *feed*

---

## Description

calculates feed demand by animal type out of a gdx file

## Usage

```
feed(
  gdx,
  file = NULL,
  level = "reg",
  detail = TRUE,
  nutrient = "dm",
  balanceflow = TRUE
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `detail` | if FALSE, only total feed demand per animal type is calculated without details on the type of feed |
| `nutrient` | The nutrient in which the results shall be calculated |
| `balanceflow` | If true, feed includes the calibration balanceflow |

## Value

feed demand by animal type as MAgPIE object (unit depends on selected nutrient attributes)

## Author(s)

Isabelle Weindl

## Examples

```
  ## Not run:
    x <- feed(gdx)

## End(Not run)
```

---

## `FoodDemandModuleConsumerPrices` *FoodDemandModuleConsumerPrices*

---

### Description

Calculates food prices that enter demand module

### Usage

```
FoodDemandModuleConsumerPrices(gdx, level = "iso")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | reg or iso |

### Value

magpie object

### Author(s)

Benjamin Leon Bodirsky

### Examples

```
  ## Not run:
    x <- FoodDemandModuleConsumerPrices(gdx)

## End(Not run)
```

---

## `FoodExpenditure` *FoodExpenditure*

---

### Description

Calculates the food expenditure in USD per year

## Usage

```
FoodExpenditure(
  gdx,
  level = "reg",
  after_shock = TRUE,
  products = "kfo",
  product_aggr = TRUE,
  per_capita = TRUE
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | spatial aggregation. can be "iso","reg","regglo","glo" |
| `after_shock` | FALSE is using the exogenous real income and the prices before a shock, TRUE is using the endogeenous real income that takes into account food price change on real income, "after_price_before_demand" takes into account price changes on real income, but assumes no demand adjustment |
| `products` | selected products or sets of products |
| `product_aggr` | if true, aggregation over products |
| `per_capita` | per capita or total population |

## Value

magpie object with per capita consumption

## Author(s)

Benjamin Leon Bodirsky

## Examples

```
  ## Not run:
    x <- FoodExpenditure(gdx)

## End(Not run)
```

---

# `FoodExpenditureShare` *FoodExpenditureShare*

---

## Description

Calculates the livestock share from the food demand model

## Usage

```
FoodExpenditureShare(
  gdx,
  level = "reg",
  after_shock = TRUE,
  products = "kfo",
  product_aggr = T
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | spatial aggregation. can be "iso","reg","regglo","glo" |
| `after_shock` | FALSE is using the exogenous real income and the prices before a shock, TRUE is using the endogeenous real income that takes into account food price change on real income |
| `products` | selected products or sets of products |
| `product_aggr` | if true, aggregation over products |

## Value

magpie object with per capita consumption

## Author(s)

Benjamin Leon Bodirsky

## Examples

```
  ## Not run:
    x <- FoodExpenditureShare(gdx)

## End(Not run)
```

---

# `foodmodelstat` *foodmodelstat*

## Description

MAgPIE food model statistics with information about convergence and number of iterations

## Usage

```
foodmodelstat(gdx, file = NULL)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |

## Value

A MAgPIE object containing number of iterations and convergence information for each time step

## Author(s)

Jan Philipp Dietrich

## Examples

```
## Not run:
x <- foodmodelstat(gdx)

## End(Not run)
```

---

# `ForestYield` *ForestYield*

## Description

calculates timber yield out of a MAgPIE gdx file

## Usage

```
ForestYield(gdx, file = NULL, level = "cell")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "cell", "reg" (regional), "glo" (global), "regglo" (regional and global) or any secdforest aggregation level defined in superAggregate |

## Details

Forest yield for timber production

## Value

Forest yield for timber production in tDM per ha per year

## Author(s)

Abhijeet Mishra, Florian Humpenoeder

## Examples

```
  ## Not run:
    x <- ForestYield(gdx)

## End(Not run)
```

---

`gdxAggregate` *gdxAggregate*

---

## Description

aggregates and disaggregates on spatial scales using mappings from the gdx files. Very specific to MAgPIE.

## Usage

```
gdxAggregate(gdx, x, weight = NULL, to, absolute = TRUE, dir = ".", ...)
```

## Arguments

| | |
|---|---|
| `gdx` | gdx file |
| `x` | object to be aggrgeagted or disaggregated |
| `weight` | weight can be either an object or a functionname in "", where the function provides the weight |
| `to` | options: grid, cell, iso, reg, glo, regglo |
| `absolute` | is it a absolute or a relative value (absolute: tons, relative: tons per hectare) |
| `dir` | for gridded outputs: magpie output directory which containts clustermap*.rds files for disaggregation. |
| `...` | further parameters handed on to weight function. |

## Value

List of magpie objects with results on country level, weight on country level, unit and description.

## Author(s)

Benjamin Leon Bodirsky, Edna J. Molina Bacca, Florian Humpenoeder

## Examples

```
## Not run:
gdp_pc <- income(gdx, level = "reg")
is.function(population)
gdp_pc_iso <- gdxAggregate(gdx = gdx, x = gdp_pc, weight = "population", to = "iso",
                           absolute = FALSE)
gdp_pc_glo <- gdxAggregate(gdx = gdx, x = gdp_pc, weight = "population", to = "glo",
                           absolute = FALSE)
gdp <- income(gdx, level = "reg", per_capita = FALSE)
gdp_iso <- gdxAggregate(gdx = gdx, x = gdp, weight = "population", to = "iso", absolute = TRUE)
gdp_glo <- gdxAggregate(gdx = gdx, x = gdp, weight = "population", to = "glo", absolute = TRUE)

## End(Not run)
```

---

# `getDirectory` *getDirectory*

## Description

support function to properly merge deprecated spamfiledirectory and dir input

## Usage

```
getDirectory(dir, spamfiledirectory)
```

## Arguments

| | |
|---|---|
| `dir` | new directory input |
| `spamfiledirectory` | old directory input |

## Value

a directory

---

# `getReport` *getReport*

## Description

Puts together a report based on a MAgPIE gdx file

## Usage

```
getReport(
  gdx,
  file = NULL,
  scenario = NULL,
  filter = c(1, 2, 7),
  detail = TRUE,
  dir = ".",
  ...
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.report. If NULL the report is returned instead as a MAgPIE object. |
| `scenario` | Name of the scenario used for the list-structure of a reporting object (x$scenario$MAgPIE). If NULL the report is returned instead as a MAgPIE object. |
| `filter` | Modelstat filter. Here you have to set the modelstat values for which results should be used. All values for time steps in which the modelstat is different or for which one of the previous modelstats were different are set to NA. |
| `detail` | Crop specific (TRUE) or aggregated outputs (FALSE) |
| `dir` | for gridded intermediate outputs: magpie output directory which contains a mapping file (rds) |
| `...` | additional arguments for write.report. Will only be taken into account if argument "file" is not NULL. |

## Details

Reports are organize with ’|’ as level delimiter and summation symbols for grouping subcategories into entities e.g. for stackplots. Notice the following hints for the summation symbol placement:

- Every name should just contain one summation symbol (mostly ’+’).
- The position of the symbol (counted in ’|’ from left side) will determine the level.
- Every subitem containing the same summation symbol in the same level with the same supercategory name will be summed.
- Items without any summation symbol will ge ignored.
- Items with different summation symbols will be summed up separately.
- In most of the cases a summation symbol will be just placed before the last level (counted in ’|’ from left side).
- It is helpful to think about which group of items should be stacked in a stackplot.

An example how a summation symbol placement could look like:

```
Toplevel
Toplevel|+|Item 1
Toplevel|+|Item 2
Toplevel|Item 2|+|Subitem 1
Toplevel|Item 2|+|Subitem 1
Toplevel|++|Item A
Toplevel|++|Item B
Toplevel|Item ?
```

## Value

A MAgPIE object containing the report in the case that "file" is NULL.

## Author(s)

Florian Humpenoeder

## Examples

```
## Not run:
x <- getReport(gdx)

## End(Not run)
```

---

# `getReportAgMIP` *getReportAgMIP*

## Description

Puts together a report for the Agricultural Model Intercom- parison and Improvement Project (AgMIP) based on a MAgPIE gdx file

## Usage

```
getReportAgMIP(
  gdx,
  file = NULL,
  scenario = NULL,
  filter = c(1, 2, 7),
  detail = TRUE,
  dir = ".",
  ...
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.report. If NULL the report is returned instead as a MAgPIE object. |
| `scenario` | Name of the scenario used for the list-structure of a reporting object (x$scenario$MAgPIE). If NULL the report is returned instead as a MAgPIE object. |
| `filter` | Modelstat filter. Here you have to set the modelstat values for which results should be used. All values for time steps in which the modelstat is different or for which one of the previous modelstats were different are set to NA. |
| `detail` | Crop specific (TRUE) or aggregated outputs (FALSE) |
| `dir` | for gridded intermediate outputs: magpie output directory which contains a mapping file (rds) |
| `...` | additional arguments for write.report. Will only be taken into account if argument "file" is not NULL. |

### Details

Reports are organize with '|' as level delimiter and summation symbols for grouping subcategories into entities e.g. for stackplots. Notice the following hints for the summation symbol placement:

- Every name should just contain one summation symbol (mostly '+').
- The position of the symbol (counted in '|' from left side) will determine the level.
- Every subitem containing the same summation symbol in the same level with the same supercategory name will be summed.
- Items without any summation symbol will ge ignored.
- Items with different summation symbols will be summed up separately.
- In most of the cases a summation symbol will be just placed before the last level (counted in '|' from left side).
- It is helpful to think about which group of items should be stacked in a stackplot.

An example how a summation symbol placement could look like:

```
Toplevel
Toplevel|+|Item 1
Toplevel|+|Item 2
Toplevel|Item 2|+|Subitem 1
Toplevel|Item 2|+|Subitem 1
Toplevel|++|Item A
Toplevel|++|Item B
Toplevel|Item ?
```

### Value

A MAgPIE object containing the report in the case that "file" is NULL.

### Author(s)

Florian Humpenoeder, Isabelle Weindl, Felicitas Beier

### Examples

```
## Not run:
x <- getReportAgMIP(gdx)

## End(Not run)
```

---

## `getReportDemandStandalone` *getReportDemandStandalone*

---

### Description

Puts together a report based on a MAgPIE gdx file

### Usage

```
getReportDemandStandalone(
  gdx,
  file = NULL,
  scenario = NULL,
  detail = FALSE,
  ...
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.report. If NULL the report is returned instead as a MAgPIE object. |
| `scenario` | Name of the scenario used for the list-structure of a reporting object (x$scenario$MAgPIE). If NULL the report is returned instead as a MAgPIE object. |
| `detail` | Crop specific (TRUE) or aggregated outputs (FALSE) |
| `...` | additional arguments for write.report. Will only be taken into account if argument "file" is not NULL. |

### Value

A MAgPIE object containing the report in the case that "file" is NULL.

### Author(s)

Florian Humpenoeder

## Examples

```
  ## Not run:
    x <- getReportDemandStandalone(gdx)

## End(Not run)
```

---

`getReportDietaryIndicators`

# *getReportDietaryIndicators*

---

## Description

reports dietary indicators on the country level. These are formatted as data.frames describing: 1. population, anthropometrics, and intake 2. caloric intake by food category (without food waste)

## Usage

```
getReportDietaryIndicators(gdx, scenario)
```

## Arguments

| | |
|---|---|
| `gdx` | filepath of the GDX file |
| `scenario` | character string describing the scenario configuration |

## Value

list of data.frames for the dietary indicators

## Author(s)

Michael Crawford, Felicitas Beier, Benjamin Bodirsky

## Examples

```
  ## Not run:
    getReportDietaryIndicators(gdx, scenario)

## End(Not run)
```

---

## getReportFableScenathon

*getReportFableScenathon*

### Description

Collects outputs from MAgPIE runs for FABLE Scenathon.

### Usage

```
getReportFableScenathon(gdx, file = NULL, iso = NULL)
```

### Arguments

| | |
|---|---|
| `gdx` | a GDX file |
| `file` | a file name the output should be written to using write.report. If ‘NULL‘ the report is returned instead as a MAgPIE object. For the easier reporting in Scenathon tabs, a .csv file extension is recommenended. |
| `iso` | country/region selection. Default ‘NULL‘, i.e. all ‘regglo‘ reporting |

### Author(s)

Miodrag Stevanovic

### Examples

```
  ## Not run:
    x <- getReportFableScenathon(gdx, file = "magpie2scenathon.csv", iso = "IND")

## End(Not run)
```

---

## getReportFSECAlessandroPassaro

*getReportFSECAlessandroPassaro*

### Description

Collects reports for Alessandro Passaro’s analysis

### Usage

```
getReportFSECAlessandroPassaro(
  magpieOutputDir,
  reportOutputDir = NULL,
  scenario = NULL
)
```

### Arguments

| | |
|---|---|
| `magpieOutputDir` | a magpie output directory which contains all the files associate with the given scenario |
| `reportOutputDir` | a folder name for the output to be written to. If NULL the report is not saved to disk, and only returned to the calling function. |
| `scenario` | the name of the scenario used. If NULL the report is not saved to disk, and only returned to the calling function. |

### Value

A list of reports

### Author(s)

Michael Crawford

### Examples

```
  ## Not run:
    x <- getReportFSECAlessandroPassaro(magpieOutputDir)

## End(Not run)
```

---

## `getReportFSECCosts` *getReportFSECCosts*

### Description

Reports cost indicators for the FSEC project

### Usage

```
getReportFSECCosts(gdx, reportOutputDir = NULL, scenario = NULL)
```

### Arguments

| | |
|---|---|
| `gdx` | a GDX file |
| `reportOutputDir` | a folder name for the output to be written to. If NULL the report is not saved to disk, and only returned to the calling function. |
| `scenario` | the name of the scenario used. If NULL the report is not saved to disk, and only returned to the calling function. |

### Value

A .csv containing the summed output of reportCostsAccounting on the region level

### Author(s)

Michael Crawford

### Examples

```
  ## Not run:
    x <- getReportFSECCosts(gdx)

## End(Not run)
```

---

## getReportFSECCropDiversityGrid
*getReportFSECCropDiversityGrid*

### Description

Reports grid cell level crop diversity for the FSEC project

### Usage

```
getReportFSECCropDiversityGrid(
  gdx,
  reportOutputDir = NULL,
  magpieOutputDir,
  scenario = NULL
)
```

### Arguments

| | |
|---|---|
| `gdx` | a GDX file |
| `reportOutputDir` | a folder name for the output to be written to. If NULL the report is not saved to disk, and only returned to the calling function. |
| `magpieOutputDir` | a magpie output directory which contains a mapping file (clustermap*.rds) for the disaggregation of grid output |
| `scenario` | the name of the scenario used. If NULL the report is not saved to disk, and only returned to the calling function. |

### Value

A list of MAgPIE objects containing the reports

## Author(s)

Patrick v. Jeetze

## Examples

```
## Not run:
x <- getReportFSECCropDiversityGrid(gdx, magpieOutputDir)

## End(Not run)
```

---

# getReportFSECPollution *getReportFSECPollution*

## Description

Reports nutrient surplus indicators for the FSEC project

## Usage

```
getReportFSECPollution(
  magpieOutputDir,
  reportOutputDir = NULL,
  scenario = NULL
)
```

## Arguments

| | |
|---|---|
| `magpieOutputDir` | a magpie output directory which contains a mapping file (clustermap*.rds) for the disaggregation of grid output |
| `reportOutputDir` | a folder name for the output to be written to. If NULL the report is not saved to disk, and only returned to the calling function. |
| `scenario` | the name of the scenario used. If NULL the report is not saved to disk, and only returned to the calling function. |

## Value

A list of MAgPIE objects containing the reports

## Author(s)

Michael Crawford

## Examples

```
  ## Not run:
    x <- getReportFSECPollution(gdx, magpieOutputDir)

## End(Not run)
```

---

`getReportFSECSimonDietz`

# *getReportFSECSimonDietz*

## Description

Collects reports for Simon Dietz' social welfare function analysis

## Usage

```
getReportFSECSimonDietz(
  magpieOutputDir,
  reportOutputDir = NULL,
  scenario = NULL
)
```

## Arguments

| | |
|---|---|
| `magpieOutputDir` | a magpie output directory which contains all the files associate with the given scenario |
| `reportOutputDir` | a folder name for the output to be written to. If NULL the report is not saved to disk, and only returned to the calling function. |
| `scenario` | the name of the scenario used. If NULL the report is not saved to disk, and only returned to the calling function. |

## Value

A list of reports

## Author(s)

Michael Crawford

## Examples

```
  ## Not run:
    x <- getReportFSECSimonDietz(magpieOutputDir)

## End(Not run)
```

---

`getReportFSECStevenLord` *getReportFSECStevenLord*

---

## Description

Collects reports for Steven Lord's cost of action / cost of inaction analysis.

## Usage

```
getReportFSECStevenLord(magpieOutputDir, reportOutputDir, scenario)
```

## Arguments

| | |
|---|---|
| `magpieOutputDir` | a magpie output directory which contains a mapping file (clustermap*.rds) for the disaggregation of grid output |
| `reportOutputDir` | a folder name for the output to be written to. If NULL the report is not saved to disk, and only returned to the calling function. |
| `scenario` | the name of the scenario used. If NULL the report is not saved to disk, and only returned to the calling function. |

## Author(s)

Michael Crawford

## Examples

```
  ## Not run:
    x <- getReportFSECStevenLord(magpieOutputDir)

## End(Not run)
```

---

`getReportGridEmissions` *getReportGridEmissions*

---

## Description

Reports all grid-level emissions available for a magpie scenario

## Usage

```
getReportGridEmissions(
  magpieOutputDir,
  reportOutputDir = NULL,
  scenario = NULL
)
```

## Arguments

| | |
|---|---|
| `magpieOutputDir` | a magpie output directory which contains a mapping file (clustermap*.rds) for the disaggregation of grid output |
| `reportOutputDir` | a folder name for the output to be written to. If NULL the report is not saved to disk, and only returned to the calling function. |
| `scenario` | the name of the scenario used. If NULL the report is not saved to disk, and only returned to the calling function. |

## Value

A list of MAgPIE objects containing the reports

## Author(s)

Michael Crawford

## Examples

```
  ## Not run:
    x <- getReportGridEmissions(gdx, magpieOutputDir)

## End(Not run)
```

---

# `getReportGridINMS` *getReportGridINMS*

## Description

Generates and saves a list of reports relevant to the INMS context

### Usage

```
getReportGridINMS(
  gdx,
  reportOutputDir = NULL,
  magpieOutputDir,
  scenario = NULL,
  filter = c(2, 7),
  version = "v13"
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `reportOutputDir` | Directory in which the reports are to be saved. If NULL, a list of reports (MAgPIE objects) is returned instead |
| `magpieOutputDir` | Directory containing the MAgPIE run which is to be processed |
| `scenario` | Name of the scenario used for the list-structure of a reporting object (x$scenario$MAgPIE). If NULL a list of reports ( MAgPIE objects) is returned instead. |
| `filter` | Modelstat filter. Here you have to set the modelstat values for which results should be used. All values for time steps in which the modelstat is different or for which one of the previous modelstats were different are set to NA. |
| `version` | Version number for this analysis |

### Value

A list of reports (MAgPIE objects)

### Author(s)

Benjamin Leon Bodirsky, Florian Humpenoeder, Michael Crawford

### Examples

```
## Not run:
x <- getReportGridINMS(gdx)

## End(Not run)
```

---

`getReportGridNitrogenPollution`

*getReportGridNitrogenPollution*

---

## Description

Reports nutrient surplus indicators as well as exceedance of the critical nitrogen surplus at the grid level

## Usage

```
getReportGridNitrogenPollution(
  magpieOutputDir,
  reportOutputDir = NULL,
  scenario = NULL
)
```

## Arguments

| | |
|---|---|
| `magpieOutputDir` | a magpie output directory which contains a mapping file (clustermap*.rds) for the disaggregation of grid output |
| `reportOutputDir` | a folder name for the output to be written to. If NULL the report is not saved to disk, and only returned to the calling function. |
| `scenario` | the name of the scenario used. If NULL the report is not saved to disk, and only returned to the calling function. |

## Value

A list of MAgPIE objects containing the reports

## Author(s)

Michael Crawford

## Examples

```
  ## Not run:
    x <- getReportGridNitrogenPollution(gdx, magpieOutputDir)

## End(Not run)
```

---

`getReportINMS` *getReportINMS*

---

## Description

Puts together a report for the INMS project based on a MAgPIE gdx file

## Usage

```
getReportINMS(
  gdx,
  file = NULL,
  scenario = NULL,
  filter = c(2, 7),
  detail = TRUE,
  dir = ".",
  ...
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.report. If NULL the report is returned instead as a MAgPIE object. |
| `scenario` | Name of the scenario used for the list-structure of a reporting object (x$scenario$MAgPIE). If NULL the report is returned instead as a MAgPIE object. |
| `filter` | Modelstat filter. Here you have to set the modelstat values for which results should be used. All values for time steps in which the modelstat is different or for which one of the previous modelstats were different are set to NA. |
| `detail` | Crop specific (TRUE) or aggregated outputs (FALSE) |
| `dir` | directory with spamfiles |
| `...` | additional arguments for write.report. Will only be taken into account if argument "file" is not NULL. |

## Value

A MAgPIE object containing the report in the case that "file" is NULL.

## Author(s)

Benjamin Bodirsky, Florian Humpenoeder

## Examples

```
## Not run:
x <- getReport(gdx)

## End(Not run)
```

---

## getReportIso *getReportIso*

## Description

Puts together a report based on a MAgPIE gdx file

## Usage

```
getReportIso(
  gdx,
  file = NULL,
  scenario = NULL,
  filter = c(1, 2, 7),
  detail = FALSE,
  dir = ".",
  ...
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.report. If NULL the report is returned instead as a MAgPIE object. |
| `scenario` | Name of the scenario used for the list-structure of a reporting object (x$scenario$MAgPIE). If NULL the report is returned instead as a MAgPIE object. |
| `filter` | Modelstat filter. Here you have to set the modelstat values for which results should be used. All values for time steps in which the modelstat is different or for which one of the previous modelstats were different are set to NA. |
| `detail` | Crop specific (TRUE) or aggregated outputs (FALSE) |
| `dir` | magpie output directory which contains a mapping file (rds) for disaggregation |
| `...` | additional arguments for write.report. Will only be taken into account if argument "file" is not NULL. |

## Details

Reports are organize with 'l' as level delimiter and summation symbols for grouping subcategories into entities e.g. for stackplots. Notice the following hints for the summation symbol placement:

- Every name should just contain one summation symbol (mostly '+').
- The position of the symbol (counted in 'l' from left side) will determine the level.
- Every subitem containing the same summation symbol in the same level with the same supercategory name will be summed.
- Items without any summation symbol will ge ignored.
- Items with different summation symbols will be summed up separately.
- In most of the cases a summation symbol will be just placed before the last level (counted in 'l' from left side).
- It is helpful to think about which group of items should be stacked in a stackplot.

An example how a summation symbol placement could look like:

```
Toplevel
Toplevel|+|Item 1
Toplevel|+|Item 2
Toplevel|Item 2|+|Subitem 1
Toplevel|Item 2|+|Subitem 1
Toplevel|++|Item A
Toplevel|++|Item B
Toplevel|Item ?
```

## Value

A MAgPIE object containing the report in the case that "file" is NULL.

## Author(s)

Florian Humpenoeder

## Examples

```
## Not run:
x <- getReport(gdx)

## End(Not run)
```

---

## `getReportMAgPIE2GAINS` *getReportMAgPIE2GAINS*

---

### Description

Puts together a report for the IIASA GAINS model based on a MAgPIE gdx file

### Usage

```
getReportMAgPIE2GAINS(
  gdx,
  folder = NULL,
  scenario = NULL,
  filter = c(2, 7),
  dir = ".",
  ...
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `folder` | a folder name the output should be written to using write.report. If NULL the report is returned instead as a MAgPIE object. |
| `scenario` | Name of the scenario used for the list-structure of a reporting object (x$scenario$MAgPIE). If NULL the report is returned instead as a MAgPIE object. |
| `filter` | Modelstat filter. Here you have to set the modelstat values for which results should be used. All values for time steps in which the modelstat is different or for which one of the previous modelstats were different are set to NA. |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |
| `...` | additional arguments for write.report. Will only be taken into account if argument "file" is not NULL. |

### Value

A MAgPIE object containing the report in the case that "file" is NULL.

### Author(s)

Benjamin Leon Bodirsky, Florian Humpenoeder

---

`getReportMAgPIE2LPJmL` *getReportMAgPIE2LPJmL*

---

## Description

Puts together a report for LPJmL or other biophysical models based on a MAgPIE gdx file

## Usage

```
getReportMAgPIE2LPJmL(
  gdx,
  folder = NULL,
  scenario = NULL,
  filter = c(2, 7),
  dir = ".",
  spamfiledirectory = "",
  ...
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `folder` | a folder name the output should be written to using write.report. If NULL the report is returned instead as a MAgPIE object. |
| `scenario` | Name of the scenario used for the list-structure of a reporting object (x$scenario$MAgPIE). If NULL the report is returned instead as a MAgPIE object. |
| `filter` | Modelstat filter. Here you have to set the modelstat values for which results should be used. All values for time steps in which the modelstat is different or for which one of the previous modelstats were different are set to NA. |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |
| `spamfiledirectory` | deprecated. please use `dir` instead |
| `...` | additional arguments for write.report. Will only be taken into account if argument "file" is not NULL. |

## Value

A MAgPIE object containing the report in the case that "file" is NULL.

## Author(s)

Benjamin Leon Bodirsky, Florian Humpenoeder

## Examples

```
## Not run:
x <- getReportMAgPIE2LPJmL(gdx)

## End(Not run)
```

---

`getReportMAgPIE2REMIND`

# *getReportMAgPIE2REMIND*

---

## Description

Based on a MAgPIE gdx file, a report is generated containing only the variables relevant for the coupling with REMIND. Basically a copy of getReport, but calling less 'reportXY()' functions.

## Usage

```
getReportMAgPIE2REMIND(gdx, file = NULL, scenario = NULL)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | A file name the output should be written to using write.report. If NULL the report is returned instead as a MAgPIE object. |
| `scenario` | Name of the scenario used for the list-structure of a reporting object (x$scenario$MAgPIE). If NULL the report is returned instead as a MAgPIE object. |

## Details

Reports are organized with '|' as level delimiter and summation symbols for grouping subcategories into entities e.g. for stackplots. Notice the following hints for the summation symbol placement:

- Every name should just contain one summation symbol (mostly '+').
- The position of the symbol (counted in '|' from left side) will determine the level.
- Every subitem containing the same summation symbol in the same level with the same supercategory name will be summed.
- Items without any summation symbol will be silently ignored.
- Items with different summation symbols will be summed up separately.
- In most of the cases a summation symbol will be just placed before the last level (counted in '|' from left side).
- It is helpful to think about which group of items should be stacked in a stackplot.

An example how a summation symbol placement could look like:

```
Toplevel
Toplevel|+|Item 1
Toplevel|+|Item 2
Toplevel|Item 2|+|Subitem 1
Toplevel|Item 2|+|Subitem 1
Toplevel|++|Item A
Toplevel|++|Item B
Toplevel|Item ?
```

## Value

A MAgPIE object containing the report.

## Author(s)

Florian Humpenoeder, David Klein

## Examples

```
## Not run:
x <- getReportMAgPIE2REMIND(gdx)

## End(Not run)
```

---

# `getReportPBindicators` *getReportPBindicators*

## Description

Puts together all reporting variables for planetary boundary indicators of MAgPIE

## Usage

```
getReportPBindicators(
  gdx,
  file = NULL,
  scenario = NULL,
  filter = c(1, 2, 7),
  dir = ".",
  ...
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.report. If NULL the report is returned instead as a MAgPIE object. |
| `scenario` | Name of the scenario used for the list-structure of a reporting object (x$scenario$MAgPIE). If NULL the report is returned instead as a MAgPIE object. |
| `filter` | Modelstat filter. Here you have to set the modelstat values for which results should be used. All values for time steps in which the modelstat is different or for which one of the previous modelstats were different are set to NA. |
| `dir` | for gridded intermediate outputs: magpie output directory which contains a mapping file (rds) |
| `...` | additional arguments for write.report. Will only be taken into account if argument "file" is not NULL. |

## Details

Reports are organize with ’|’ as level delimiter and summation symbols for grouping subcategories into entities e.g. for stackplots. Notice the following hints for the summation symbol placement:

- Every name should just contain one summation symbol (mostly ’+’).
- The position of the symbol (counted in ’|’ from left side) will determine the level.
- Every subitem containing the same summation symbol in the same level with the same supercategory name will be summed.
- Items without any summation symbol will ge ignored.
- Items with different summation symbols will be summed up separately.
- In most of the cases a summation symbol will be just placed before the last level (counted in ’|’ from left side).
- It is helpful to think about which group of items should be stacked in a stackplot.

An example how a summation symbol placement could look like:

```
Toplevel
Toplevel|+|Item 1
Toplevel|+|Item 2
Toplevel|Item 2|+|Subitem 1
Toplevel|Item 2|+|Subitem 1
Toplevel|++|Item A
Toplevel|++|Item B
Toplevel|Item ?
```

## Value

A MAgPIE object containing the report in the case that "file" is NULL.

## Author(s)

Felicitas Beier

## Examples

```
## Not run:
x <- getReport(gdx)

## End(Not run)
```

---

## `grassyld` *grassland yields*

### Description

Calculates grassland yields based on a MAgPIE gdx file

### Usage

```
grassyld(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

A MAgPIE object containing grassland yields values

### Author(s)

Marcos Alves

### Examples

```
## Not run:
x <- grassyld(gdx)

## End(Not run)
```

---

`GrowingStock` *GrowingStock*

---

## Description

reads woody growing stock out of a MAgPIE gdx file

## Usage

```
GrowingStock(gdx, file = NULL, level = "regglo", indicator = "relative")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "cell", "reg" (regional), "glo" (global), "regglo" (regional and global) or any secdforest aggregation level defined in superAggregate |
| `indicator` | If the reported numbers are relative (mio m3/ha) or absolute (mio. m3). Default is relative. |

## Details

Growing stock for producing woody materials consist of growing stock from plantations (forestry), secondary and primary forest as well as other land (natveg)

## Value

Growing stock in m3 per ha

## Author(s)

Abhijeet Mishra

## Examples

```
## Not run:
x <- GrowingStock(gdx)

## End(Not run)
```

---

## `harvested_area_timber` *harvested_area_timber*

---

### Description

Reads wood harvest area separated by source (primforest, secdforest, forestry, other) and age classes from a gdx. The data is on cluster level and the unit is Mha per year.

### Usage

```
harvested_area_timber(
  gdx,
  file = NULL,
  level = "cell",
  aggregateAgeClasses = TRUE
)
```

### Arguments

| | |
|---|---|
| `gdx` | A fulldata.gdx of a magpie run, usually with endogenous forestry enabled |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "cell", "reg" (regional), "glo" (global), "regglo" (regional and global) or any secdforest aggregation level defined in superAggregate |
| `aggregateAgeClasses` | If TRUE, age classes are aggregated |

### Value

Area harvested for wood in Mha per year as a magpie object

### Author(s)

Abhijeet Mishra, Pascal Sauer

---

## `hourlyLaborCosts` *hourlyLaborCosts*

---

### Description

returns hourly labor costs in agriculture from MAgPIE results

### Usage

```
hourlyLaborCosts(gdx, level = "reg", file = NULL)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | spatial aggregation to report employment ("iso", "reg", "glo", or "regglo") |
| `file` | a file name the output should be written to using write.magpie |

## Value

hourly labor costs in agriculture

## Author(s)

Debbora Leip

## Examples

```
## Not run:
x <- hourlyLaborCosts(gdx)

## End(Not run)
```

---

# `Hunger` *Hunger*

## Description

Calculates the share of people living in hunger.

## Usage

```
Hunger(
  gdx,
  level = "reg",
  after_shock = TRUE,
  calibrated = FALSE,
  share = TRUE
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | spatial aggregation. can be "iso","reg","regglo","glo" |
| `after_shock` | FALSE is using the exogenous real income and the prices before a shock, TRUE is using the endogeenous real income that takes into account food price change on real income |
| `calibrated` | if calibrated is TRUE, kcal values are calibrated to better match historical years |
| `share` | share of population that is undernourished |

## Value

magpie object with hunger (mio people) or hunger share

## Author(s)

Benjamin Leon Bodirsky

## Examples

```
  ## Not run: 
    x <- Hunger(gdx)
  
## End(Not run)
```

---

# income *income*

## Description

Calculates income based on a MAgPIE gdx file

## Usage

```
income(
  gdx,
  file = NULL,
  level = "reg",
  per_capita = TRUE,
  type = "ppp",
  after_shock = FALSE
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `per_capita` | income per capita or aggregated for the total population |
| `type` | ppp for purchase power parity, mer for market exchange rate |
| `after_shock` | FALSE is using the exogenous real income, TRUE is using the endogeenous real income that takes into account food price change on real income |

### Value

annual income as MAgPIE object (unit depends on per_capita: US$2017 MER/cap/yr (TRUE), US$2017 MER/yr (FALSE))

### Author(s)

Florian Humpenoeder, Benjamin Bodirsky, Felicitas Beier

### Examples

```
## Not run:
x <- income(gdx)

## End(Not run)
```

---

## `Intake` *Intake*

### Description

Calculates the per-capita kcal intake from the food demand model

### Usage

```
Intake(
  gdx,
  file = NULL,
  level = "reg",
  calibrated = TRUE,
  pregnancy = FALSE,
  per_capita = TRUE,
  age = FALSE,
  sex = FALSE,
  bmi_groups = FALSE,
  dir = ".",
  spamfiledirectory = ""
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "iso" ISO country codes, "reg" (regional), "glo" (global) |
| `calibrated` | if FALSE, the true regression outputs are used, if TRUE the values calibrated to the start years are used |

| | |
|---|---|
| `pregnancy` | if TRUE, adding the intake requirements for lactation and pregnancy |
| `per_capita` | per capita or aggregated for the population |
| `age` | if FALSE age and sex is aggregated |
| `sex` | if TRUE, data is provided by sex |
| `bmi_groups` | if TRUE data is proided by BMI group |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |
| `spamfiledirectory` | deprecated. please use `dir` instead |

## Details

Demand definitions are equivalent to FAO Food supply categories

## Value

calories as MAgPIE object (unit depends on per_capita: kcal/cap/day (TRUE), kcal/day (FALSE))

## Author(s)

Benjamin Leon Bodirsky

## Examples

```
  ## Not run:
    x <- Intake(gdx)

## End(Not run)
```

---

# `IntakeDetailed` *IntakeDetailed*

## Description

Calculates detailed or aggregated per-capita kcal intake including exogenous scenarios

## Usage

```
IntakeDetailed(
  gdx,
  file = NULL,
  level = "reg",
  product_aggr = FALSE,
  dir = ".",
  spamfiledirectory = ""
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `product_aggr` | aggregate over products or not (boolean) |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |
| `spamfiledirectory` | deprecated. please use `dir` instead |

### Details

Calculation of kcal food intake is possible for both exogenous diet scenarios and endogenous estimation from food demand model

### Value

Calories as MAgPIE object (unit: kcal/cap/day)

### Author(s)

Isabelle Weindl

### Examples

```
  ## Not run:
    x <- IntakeDetailed(gdx)

## End(Not run)
```

---

## `IntakeDetailedProtein` *IntakeDetailedProtein*

### Description

Calculates food-specific per-capita protein intake from magpie results in grams.

### Usage

```
IntakeDetailedProtein(
  gdx,
  file = NULL,
  level = "reg",
  product_aggr = FALSE,
  dir = "."
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `product_aggr` | aggregate over products or not (boolean) |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |

### Value

Protein intake as MAgPIE object (unit: grams/cap/day)

### Author(s)

Vartika Singh, Isabelle Weindl

### Examples

```
  ## Not run:
    x <- IntakeDetailedProtein(gdx)

## End(Not run)
```

---

## `Kcal` *Kcal*

### Description

Calculates the per-capita kcal consumption from the food demand model

### Usage

```
Kcal(
  gdx,
  file = NULL,
  level = "reg",
  products = "kfo",
  product_aggr = TRUE,
  after_shock = TRUE,
  calibrated = TRUE,
  magpie_input = FALSE,
  attributes = "kcal",
  per_capita = TRUE,
  dir = ".",
  spamfiledirectory = ""
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "iso" ISO country codes, "reg" (regional), "glo" (global) |
| `products` | Selection of products (either by naming products, e.g. "tece", or naming a set,e.g."kcr") |
| `product_aggr` | aggregate over products or not (boolean) |
| `after_shock` | FALSE is using the exogenous real income and the prices before a shock, TRUE is using the endogenous real income that takes into account food price change on real income |
| `calibrated` | if FALSE, the true regression outputs are used, if TRUE the values calibrated to the start years are used |
| `magpie_input` | TRUE or FALSE. This setting is only activate if arguments "calibrated" and "after_shock" are set to TRUE and else ignored. If set as TRUE, the per-capita kcal consumption values finally entering MAgPIE as input are used, which drive the behaviour of the MAgPIE model, excluding countries not listed in FAO. If set as FALSE, the per-capita kcal consumption values as calculated in the food demand model are used, including countries not listed in FAO. |
| `attributes` | unit: kilocalories per day ("kcal"), g protein per day ("protein"). Mt reactive nitrogen ("nr"). |
| `per_capita` | per capita or aggregated for the population |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |
| `spamfiledirectory` | deprecated. please use `dir` instead |

## Details

Demand definitions are equivalent to FAO Food supply categories

## Value

calories as MAgPIE object (unit depends on per_capita: kcal/cap/day (TRUE), kcal/day (FALSE))

## Author(s)

Benjamin Leon Bodirsky, Isabelle Weindl

## Examples

```
  ## Not run:
    x <- Kcal(gdx)

## End(Not run)
```

---

`laborCosts` *laborCosts*

---

## Description

reads labor costs for crop and livestock production from gdx file

## Usage

```
laborCosts(gdx, products = "kcr", file = NULL, level = "grid", dir = ".")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `products` | products for which labor costs should be reported ("kcr" or "kli", for other products use factorCosts()) |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation ("grid" or "iso", for regional/global use factorCosts()) |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |

## Value

MAgPIE object containing labor costs [million US$17]

## Author(s)

Debbora Leip

## Examples

```
## Not run:
x <- laborCosts(gdx)

## End(Not run)
```

---

`laborProductivity` *laborProductivity*

---

## Description

calculates labor productivity in crop sector (kg DM per hour) from a MAgPIE gdx file

## Usage

```
laborProductivity(gdx, level = "reg", productAggr = TRUE)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | spatial aggregation to report productivity ("cell","reg", "regglo", "glo") |
| `productAggr` | Aggregate over products or not (boolean) |

## Value

labor productivity in crop sector (kg DM per hour)

## Author(s)

Xiaoxi Wang, Ruiying Du, Debbora Leip

## Examples

```
## Not run:
x <- laborProductivity(gdx)

## End(Not run)
```

---

`land` *land*

---

## Description

reads land out of a MAgPIE gdx file

### Usage

```
land(
  gdx,
  file = NULL,
  level = "reg",
  types = NULL,
  subcategories = NULL,
  sum = FALSE,
  dir = ".",
  spamfiledirectory = ""
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in gdxAggregate |
| `types` | NULL or a vector of strings. If NULL, all land types are used. Options are "crop", "past", "forestry", "primforest","secdforest, "urban", "other", "primother" and "secdother" |
| `subcategories` | NULL or vector of strings. If NULL, no subcategories are returned. Meaningful options are "crop, "forestry" and "other" |
| `sum` | determines whether output should be land-type-specific (FALSE) or aggregated over all types (TRUE). |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |
| `spamfiledirectory` | deprecated. please use `dir` instead |

### Value

land as MAgPIE object (Mha)

### Author(s)

Jan Philipp Dietrich, Florian Humpenoeder, Benjamin Leon Bodirsky, Patrick v. Jeetze

### See Also

`reportLandUse`

### Examples

```
## Not run:
x <- land(gdx)
```

```
## End(Not run)
```

---

## `landCarbonSink` *Land Carbon Sink Adjustment Factors*

### Description

Indirect human-induced emissions in the land use system

### Usage

```
landCarbonSink(
  gdx,
  file = NULL,
  level = "reg",
  cumulative = FALSE,
  baseyear = 1995,
  source = "Grassi"
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global). |
| `cumulative` | Logical; Determines if emissions are reported annually (FALSE) or cumulative (TRUE). The starting point for cumulative emissions is y1995. |
| `baseyear` | Baseyear used for cumulative emissions (default = 1995) |
| `source` | Currently only "Grassi", which uses pre-calculated adjustment factors from Grassi et al 2021 (DOI 10.1038/s41558-021-01033-6). Can be extended in the future to also include "PIK", based on data from LPJmL. |

### Details

Calculates global and regional Land Carbon Sink Adjustment Factors

### Value

Land Carbon Sink Adjustment Factors (Mt CO2 per year or cumulative)

### Author(s)

Florian Humpenoeder

## Examples

```
## Not run:
x <- landCarbonSink(gdx)

## End(Not run)
```

---

`landForestry` *landForestry*

---

## Description

reads and compiles forestry land subcategories from a MAgPIE gdx file

## Usage

```
landForestry(gdx, file = NULL, level = "reg")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |

## Value

land as MAgPIE object (Mha)

## Author(s)

Florian Humpenoeder

## See Also

`reportLandUse`

## Examples

```
  ## Not run:
    x <- land(gdx)

## End(Not run)
```

## `land_price` *land_price*

### Description

Calculates MAgPIE MAgPIE land shadow prices based on a gdx file

### Usage

```
land_price(
  gdx,
  file = NULL,
  level = "reg",
  ignore_lowbound = FALSE,
  absolute = TRUE,
  digits = 4
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `ignore_lowbound` | Some shadow prices are positive (see Details), corresponding to a lower bound for that pool. `TRUE` sets them to 0. Default value: `FALSE`. |
| `absolute` | Should the absolute value of all the marginals be taken into calculations? `TRUE` (default) of `FALSE`. See Details. |
| `digits` | rounding accuracy for the output |

### Details

The land price is obtained through marginals of the "oq_cropland" constraint. The majority of these marginals are negative values, and a negligible number of them are positive. This is the consequence of the constraint binding either on upper or lower level. The parameter `ignore_lowbound` removes all the positive marginals from land price calculation (negligible), and parameter `absolute` transforms them into negative values (to be all together reported as positive values at the final calculation).

### Value

A MAgPIE object containing the land shadow prices (US$2017/ha).

### Author(s)

Markus Bonsch, Misko Stevanovic

## Examples

```
  ## Not run:
    x <- land_price(level="regglo", products="kcr")

## End(Not run)
```

---

## lastIter *lastIter*

## Description

Returns the value of a parameter in the last iteration

## Usage

```
lastIter(gdx, param, secondlast = FALSE)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `param` | Parameter to be returned |
| `secondlast` | if TRUE, reads the secondlast iteration. For MAgPIE results, usually there is no last iteration as the food demand model reaches convergence before MAgPIE starts. |

## Value

magpie object

## Author(s)

Benjamin Leon Bodirsky

## Examples

```
  ## Not run:
    x <- lastIter(gdx)

## End(Not run)
```

---

## `LivestockDemStructure` *LivestockDemStructure*

---

### Description

Calculates the share of different livestock commodities in total livestock product consumption on the basis of chosen attribute

### Usage

```
LivestockDemStructure(
  gdx,
  file = NULL,
  level = "reg",
  after_shock = TRUE,
  calibrated = TRUE,
  attributes = "kcal",
  fish = FALSE
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "iso" ISO country codes, "reg" (regional), "glo" (global) |
| `after_shock` | FALSE is using the exogenous real income and the prices before a shock, TRUE is using the endogeenous real income that takes into account food price change on real income |
| `calibrated` | if FALSE, the true regression outputs are used, if TRUE the values calibrated to the start years are used |
| `attributes` | unit: kilocalories per day ("kcal"), g protein per day ("protein"). Mt reactive nitrogen ("nr"). |
| `fish` | if true, livestock share includes fish, otherwhise not |

### Value

magpie object with the livestock demand structure in a region or country. Unit is dimensionless, but value depends on chosen attribute

### Author(s)

Isabelle Weindl

## Examples

```
  ## Not run:
    x <- LivestockDemStructure(gdx)

## End(Not run)
```

---

`LivestockShare` *LivestockShare*

## Description

Calculates the livestock share from the food demand model

## Usage

```
LivestockShare(
  gdx,
  file = NULL,
  level = "reg",
  after_shock = TRUE,
  calibrated = TRUE,
  attributes = "kcal",
  fish = TRUE
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "iso" ISO country codes, "reg" (regional), "glo" (global) |
| `after_shock` | FALSE is using the exogenous real income and the prices before a shock, TRUE is using the endogeenous real income that takes into account food price change on real income |
| `calibrated` | if FALSE, the true regression outputs are used, if TRUE the values calibrated to the start years are used |
| `attributes` | unit: kilocalories per day ("kcal"), g protein per day ("protein"). Mt reactive nitrogen ("nr"). |
| `fish` | if true, livestock share includes fish, otherwhise not |

## Value

magpie object with the livestock share in a region or country. Unit is dimensionsless, but value depends on chosen attribute

## Author(s)

Benjamin Bodirsky

## Examples

```
  ## Not run: 
    x <- LivestockShare(gdx)

## End(Not run)
```

---

# `malmquist` *malmquist*

## Description

calcluates malmquist index based on a MAgPIE gdx file

## Usage

```
malmquist(gdx, file = NULL)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |

## Value

A MAgPIE object containing the malmquist index

## Author(s)

Xiaoxi Wang

## Examples

```
  ## Not run: 
    x <- malmquist(gdx)

## End(Not run)
```

---

`ManureExcretion` *ManureExcretion*

---

### Description

downscales Manure Excretion

### Usage

```
ManureExcretion(
  gdx,
  level = "reg",
  products = "kli",
  awms = c("grazing", "stubble_grazing", "fuel", "confinement"),
  agg = TRUE,
  dir = "."
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | aggregation level: glo, reg, cell, grid, iso |
| `products` | livestock products |
| `awms` | large animal waste management categories: "grazing","stubble_grazing","fuel","confinement"), |
| `agg` | aggregation over "awms" or over "products". |
| `dir` | directory with spamfiles |

### Value

MAgPIE object

### Author(s)

Benjamin Leon Bodirsky

### Examples

```
  ## Not run:
    x <- ManureExcretion(gdx)

## End(Not run)
```

---

`metadata_comments` *metadata_comments*

---

## Description

set metadata comments to magpie4 objects

## Usage

```
metadata_comments(x, unit, description, comment, note)
```

## Arguments

| | |
|---|---|
| `x` | magpie object (magpie4) |
| `unit` | provide unit |
| `description` | provide short description |
| `comment` | optional comment |
| `note` | optional note |

## Value

vector of comments following order of input (unit, description, comment, note - further: origin, creation data)

## Author(s)

Benjamin Bodirsky, Jannes Breier

## Examples

```
  ## Not run:
    x <- metadata_comments(x,unit,description,comment,note)

## End(Not run)
```

---

## `modelstat` *modelstat*

---

### Description

MAgPIE model stat of all optimizations - main optimization and (if used) presolve optimization.

### Usage

```
modelstat(gdx, file = NULL)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |

### Value

A MAgPIE object containing the modelstat

### Author(s)

Jan Philipp Dietrich

### Examples

```
  ## Not run:
    x <- modelstat(gdx)

## End(Not run)
```

---

## `m_yeardiff` *m_yeardiff*

---

### Description

Calculates the parameter m_yeardiff, which is a macro within MAgPIE.

### Usage

```
m_yeardiff(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

a magpie object with the length of each timestep

## Author(s)

Benjamin Leon Bodirsky

## Examples

```
  ## Not run:
    x <- m_yeardiff(gdx)

## End(Not run)
```

---

## `NetForestChange` *NetForestChange*

## Description

Calculates net forest change based on a MAgPIE gdx file

## Usage

```
NetForestChange(gdx, file = NULL, level = "cell", lowpass = 3)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `lowpass` | number of lowpass filter iterations (default = 3) |

## Value

Net Forest Change as MAgPIE object (Mha per year)

## Author(s)

Florian Humpenoeder

## Examples

```
## Not run:
x <- NetForestChange(gdx)

## End(Not run)
```

---

## `NitrogenBudget` *NitrogenBudget*

---

### Description

calculates projections of Nitrogen Budgets for Croplands (Tg Nr per) from a MAgPIE gdx file

### Usage

```
NitrogenBudget(
  gdx,
  include_emissions = FALSE,
  level = "reg",
  dir = ".",
  debug = FALSE,
  cropTypes = FALSE
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `include_emissions` | TRUE also divides the N surplus into different emissions |
| `level` | aggregation level, reg, glo or regglo, cell, iso or grid |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |
| `debug` | debug mode TRUE makes some consistency checks between estimates for different resolutions. |
| `cropTypes` | FALSE for aggregate results; TRUE for crop-specific results |

### Author(s)

Benjamin Leon Bodirsky, Michael Crawford, Edna J. Molina Bacca, Florian Humpenoeder

### Examples

```
## Not run:
x <- NitrogenBudget(gdx)

## End(Not run)
```

---

## `NitrogenBudgetNonagland` *NitrogenBudgetNonagland*

---

### Description

calculates projections of Nitrogen Budgets for non-agricutlural land from a MAgPIE gdx file

### Usage

```
NitrogenBudgetNonagland(gdx, level = "reg", dir = ".")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | aggregation level, reg, glo or regglo |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |

### Author(s)

Benjamin Leon Bodirsky, Edna J. Molina Bacca

### Examples

```
  ## Not run:
    x <- NitrogenBudgetNonagland(gdx)

## End(Not run)
```

---

## `NitrogenBudgetPasture` *NitrogenBudgetPasture*

---

### Description

calculates projections of Nitrogen Budgets for Croplands from a MAgPIE gdx file

### Usage

```
NitrogenBudgetPasture(gdx, include_emissions = FALSE, level = "reg", dir = ".")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `include_emissions` | TRUE also divides the N surplus into different emissions |
| `level` | aggregation level, reg, glo or regglo, cell, grid, iso |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |

## Author(s)

Benjamin Leon Bodirsky, Edna J. Molina Bacca

## Examples

```
## Not run:
x <- NitrogenBudgetPasture(gdx)

## End(Not run)
```

---

`NitrogenBudgetWithdrawals`

# *NitrogenBudgetWithdrawals*

## Description

calculates projections of Nitrogen Budgets withdrawals for Croplands from a MAgPIE gdx file

## Usage

```
NitrogenBudgetWithdrawals(
  gdx,
  kcr = "sum",
  net = TRUE,
  level = "reg",
  dir = "."
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `kcr` | "sum" provides the totals over all crops, "kcr" provides outputs by kcr |
| `net` | TRUE only provides total net-withdrawals, otherwise all categories are returned (fixation and seed are returned positive, not negative) |
| `level` | aggregation level, reg, glo or regglo, cell, grid or iso |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |

## Author(s)

Benjamin Leon Bodirsky, Michael Crawford

## Examples

```
## Not run:
x <- NitrogenBudgetWithdrawals(gdx)

## End(Not run)
```

---

`out` *out*

---

## Description

Function to safely returns parameters. Function returns either the output or writes it to a file. Please use this function when you write own GDX output functions.

## Usage

```
out(x,file)
```

## Arguments

| | |
|---|---|
| `x` | an object that can be converted to a MAgPIE object |
| `file` | file name of a file it should be written to. NULL, if x should be returned instead to be written to a file. |

## Value

NULL or x as MAgPIE object

## Author(s)

Jan Philipp Dietrich

---

## `outputCheck` *outputCheck*

---

### Description

Function to check a MAgPIE gdx file for known problems (e.g. non-zero dummy variables). The function will throw warnings for problem found in the outputs.

### Usage

```
outputCheck(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Author(s)

Jan Philipp Dietrich

### Examples

```
## Not run:
outputCheck(gdx)

## End(Not run)
```

---

## `outputPerWorker` *outputPerWorker*

---

### Description

returns output per worker in crop+livestock production

### Usage

```
outputPerWorker(gdx, level = "reg", file = NULL)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | spatial aggregation to report employment ("reg", "glo", or "regglo") |
| `file` | a file name the output should be written to using write.magpie |

### Value

output per worker as magpie object

## Author(s)

Debbora Leip

## Examples

```
## Not run:
x <- outputPerWorker(gdx)

## End(Not run)
```

---

# `PeatlandArea` *PeatlandArea*

## Description

reads peatland area out of a MAgPIE gdx file

## Usage

```
PeatlandArea(gdx, file = NULL, level = "cell", sum = TRUE)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "cell", "reg" (regional), "glo" (global), "regglo" (regional and global) or any aggregation level defined in superAggregate. In addition "climate" for the 3 climate regions tropical, temperate and boreal is available. |
| `sum` | sum over land types TRUE (default) or FALSE |

## Details

Intact, degraded and rewettet peatland area

## Value

Peatland area in Mha

## Author(s)

Florian Humpenoeder

## Examples

```
  ## Not run:
    x <- PeatlandArea(gdx)

## End(Not run)
```

---

## `PeatlandEmissions` *PeatlandEmissions*

---

### Description

reads peatland GHG emissions out of a MAgPIE gdx file

### Usage

```
PeatlandEmissions(
  gdx,
  file = NULL,
  level = "cell",
  unit = "gas",
  cumulative = FALSE,
  baseyear = 1995,
  lowpass = 0,
  sum = TRUE,
  intact = FALSE
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "cell", "reg" (regional), "glo" (global), "regglo" (regional and global) or any aggregation level defined in superAggregate. In addition "climate" for the 3 climate regions tropical, temperate and boreal is available. |
| `unit` | global warming potential (gwp) or gas (gas) |
| `cumulative` | FALSE (default) or TRUE |
| `baseyear` | Baseyear used for cumulative emissions (default = 1995) |
| `lowpass` | number of lowpass filter iterations (default = 0) |
| `sum` | sum over land types TRUE (default) or FALSE |
| `intact` | report GHG emissions from intact peatlands FALSE (default) or TRUE |

### Details

Peatland GHG emissions: CO2, DOC, CH4 and N2O

### Value

Peatland GHG emissions in Mt CO2eq (if unit="gwp") or Mt of the respective gas (if unit="gas")

### Author(s)

Florian Humpenoeder

### Examples

```
  ## Not run:
    x <- PeatlandArea(gdx)

## End(Not run)
```

---

## `PlantationEstablishment` *PlantationEstablishment*

### Description

reads carbon stocks in harvested timber out of a MAgPIE gdx file

### Usage

```
PlantationEstablishment(gdx, file = NULL, level = "cell")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "cell", "reg" (regional), "glo" (global), "regglo" (regional and global) or any secdforest aggregation level defined in superAggregate |

### Details

Area newly established in current time step for future timber production

### Value

Area newly for timber production

### Author(s)

Abhijeet Mishra

### Examples

```
  ## Not run:
    x <- PlantationEstablishment(gdx)

## End(Not run)
```

## `population` *population*

### Description

reads population out of a MAgPIE gdx file

### Usage

```
population(
  gdx,
  file = NULL,
  level = "reg",
  age = FALSE,
  sex = FALSE,
  bmi_groups = FALSE,
  dir = ".",
  spamfiledirectory = ""
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `age` | if TRUE, population is split up by age groups |
| `sex` | if TRUE, population is split up by sex |
| `bmi_groups` | if TRUE, the population will be split up in body-mass-index groups. |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |
| `spamfiledirectory` | deprecated. please use `dir` instead |

### Value

population as MAgPIE object (million people)

### Author(s)

Florian Humpenoeder, Benjamin Bodirsky, Isabelle Weindl

### See Also

`reportPopulation`

## Examples

```
## Not run:
x <- population(gdx)

## End(Not run)
```

---

`PriceElasticities` *PriceElasticities*

## Description

Calculates the physical elasticity for food demand

## Usage

```
PriceElasticities(
  gdx,
  file = NULL,
  level = "reg",
  calibrated = TRUE,
  products = "kfo"
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "iso" ISO country codes, "reg" (regional), "glo" (global) |
| `calibrated` | if FALSE, the true regression outputs are used, if TRUE the values calibrated to the start years are used |
| `products` | set of the products for which the elasticity should be estimated. Please note that this stills remains an elasticity relative to total food expenditure. So its the change in consumption of one good when the prices of all products change according to the scenario. |

## Value

magpie object with the livestock share in a region or country. Unit is dimensionsless, but value depends on chosen attribute

## Author(s)

Benjamin Bodirsky

## Examples

```
  ## Not run:
    x <- PriceElasticities(gdx)

## End(Not run)
```

---

# PriceGHG *PriceGHG*

## Description

reads GHG emission prices out of a MAgPIE gdx file

## Usage

```
PriceGHG(gdx, file = NULL, level = "reg", aggr = "max")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `aggr` | aggregation used for global value; "max" (maxium value accross regions) or "weight" (weighted by population) |

## Value

GHG emission prices as MAgPIE object (US$2017/tCO2, US$2017/tN2O, US$2017/tCH4)

## Author(s)

Florian Humpenoeder, Amsalu W. Yalew

## See Also

`reportPriceGHG`

## Examples

```
  ## Not run:
    x <- PriceGHG(gdx)

## End(Not run)
```

---

`priceIndex` *priceIndex*

---

## Description

calcluates price indicies based on a MAgPIE gdx file

## Usage

```
priceIndex(
  gdx,
  file = NULL,
  level = "reg",
  products = "kall",
  index = "lasp",
  chain = FALSE,
  baseyear = "y2005",
  round = TRUE,
  type = "consumer",
  product_aggr = TRUE
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `products` | Selection of products (either by naming products, e.g. "tece", or naming a set,e.g."kcr") |
| `index` | "lasp" (Laspeyres-Index: baseyear weighting), "paas" (Paasche-Index: current weighting), "fish" (Fisher-Index: geometric mean of "lasp" and "paas") |
| `chain` | Chain Index: if true, the base period for each time period is the immediately preceding time period. Can be combined with all of the above indices |
| `baseyear` | baseyear of the price index |
| `round` | shall the results be rounded? |
| `type` | For whom are the prices important? "producer" are the prices that farmer face, as they also produce intermediate products (seed, feed). "consumer" are the prices for the end consumer faces (supermarket, bioenergy plant). Currently, the only difference is the basket composition (ideally, also prices should differ between regions) |
| `product_aggr` | aggregate over products or not (boolean) |

### Value

A MAgPIE object containing price indices for consumers or producers (depending on type)

### Author(s)

Jan Philipp Dietrich, Florian Humpenoeder, Benjamin Bodirsky

### Examples

```
  ## Not run:
    x <- priceIndex(gdx)

## End(Not run)
```

---

## `priceIndexFood` *priceIndexFood*

### Description

calcluates price indicies based on a MAgPIE gdx file

### Usage

```
priceIndexFood(
  gdx,
  file = NULL,
  level = "reg",
  index = "lasp",
  chain = FALSE,
  baseyear = "y2005",
  round = TRUE,
  product_aggr = TRUE
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `index` | "lasp" (Laspeyres-Index: baseyear weighting), "paas" (Paasche-Index: current weighting), "fish" (Fisher-Index: geometric mean of "lasp" and "paas") |
| `chain` | Chain Index: if true, the base period for each time period is the immediately preceding time period. Can be combined with all of the above indices |

| | |
|---|---|
| `baseyear` | baseyear of the price index. type model to take baseyear 2010 with literature prices |
| `round` | shall the results be rounded? |
| `product_aggr` | aggregate over products or not (boolean) |

### Value

A MAgPIE object containing price indices for consumers or producers (depending on type)

### Author(s)

Jan Philipp Dietrich, Florian Humpenoeder, Benjamin Bodirsky

### Examples

```
  ## Not run:
    x <- priceIndexFood(gdx)

## End(Not run)
```

---

## `prices` *prices*

### Description

calcluates prices based on a MAgPIE gdx file

### Usage

```
prices(
  gdx,
  file = NULL,
  level = "reg",
  products = "kall",
  product_aggr = FALSE,
  attributes = "dm",
  type = "consumer",
  glo_weight = "production"
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |

| | |
|---|---|
| `products` | Selection of products (either by naming products, e.g. "tece", or naming a set,e.g."kcr") |
| `product_aggr` | aggregate over products or not (boolean) |
| `attributes` | USD05MER per ton X (dm,nr,p,k,wm) except gross energy (ge) where it is USD05MER per GJ |
| `type` | "consumer" or "producer" prices. Producers' prices are calculated on the regional level as a sum of regional trade equation marginal values and respective global trade equation marginal values.For the non traded commodities, both global and regional producers prices are set to zero instead of NaN. |
| `glo_weight` | Decides the calculation of global prices. Weighting schemes are applied for estimation of global producer price. If `"export"` prices are calculated as average of regional exporters' prices, weighted by the export volumes. If `"production"` (default), prices are calculated as average of regional prices weighted by regional production. If `"free_trade"`, the global prices are directly taken from the shadow prices of the global trade constraint, and no averaging is performed. |

## Value

A MAgPIE object containing the consumer's or producers' prices (unit depends on attributes)

## Author(s)

Misko Stevanovic, Florian Humpenoeder, Jan Philipp Dietrich, Xiaoxi Wang, Edna J. Molina Bacca

## Examples

```
## Not run:
x <- prices(gdx)

## End(Not run)
```

---

# `PrimSecdOtherLand` *PrimSecdOtherLand*

## Description

Calculates share of primary and secondary non-forest vegetation for different aggregation levels based on gridded magpie output and initial shares of primary and secondary non-forest vegetation.

## Usage

```
PrimSecdOtherLand(
  x,
  ini_file,
  ini_year = "y1995",
  file = NULL,
```

```
  level = "grid",
  unit = "Mha"
)
```

### Arguments

| | |
|---|---|
| `x` | Time series of land pools (model output) containing only one aggregated class for other land. Can be a file or magclass object. |
| `ini_file` | Initialisation file for primary and secondary other land (e.g. based on 1995 MAgPIE land-use initialisation values). Must have the same spatial resolution as `x`. |
| `ini_year` | Reference year for estimating primary and secondary other land shares, must be included in `ini_file`. |
| `file` | a file name the output should be written to using `write.magpie` |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate. The unit of output for the cases above is Mha. If level "grid" is specified the unit of output can be chosen between "share" and "Mha". |
| `unit` | "Mha" or "share". Defines the unit of the gridded output, see also `level`. |

### Value

`x` including land area for primary and secondary non-forested vegetation in MAgPIE (other land) as MAgPIE object; either as unit of area (Mha) or as fraction of total land per grid cell (share).

### Author(s)

Patrick v. Jeetze, Kristine Karstens

### Examples

```
## Not run:
x <- "./cell.land_0.5.nc"
land <- PrimSecdOtherLand(x)

# direct use of disaggregation output
land <- PrimSecdOtherLand(land_hr)

## End(Not run)
```

---

`processing` *processing*

---

## Description

Calculates MAgPIE disaggregated processing out of a gdx file

## Usage

```
processing(
  gdx,
  level = "reg",
  product_aggr = FALSE,
  attributes = "dm",
  type = NULL,
  indicator = "secondary_from_primary"
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | Level of regional aggregation ("reg", "glo", "regglo") |
| `product_aggr` | aggregate over products or not (boolean) |
| `attributes` | dry matter: Mt ("dm"), gross energy: PJ ("ge"), reactive nitrogen: Mt ("nr"), phosphor: Mt ("p"), potash: Mt ("k"), wet matter: Mt ("wm"). Can also be a vector. |
| `type` | Demand type(s): "Food", "Feed", "Processing", "Material", "Bioenergy", "Seed", "Supply chain loss", "Domestic Balanceflow"; NULL returns all types |
| `indicator` | process or secondary product output |

## Details

Demand definitions are equivalent to FAO CBS categories

## Value

processing as MAgPIE object (Unit depends on attributes)

## Author(s)

David Chen, Benjamin Leon Bodirsky

## Examples

```
  ## Not run:
   x <- processing(gdx = gdx, level="regglo", products="kcr", indicator="primary_to_process")

## End(Not run)
```

---

## production *production*

### Description

reads production out of a MAgPIE gdx file

### Usage

```
production(
  gdx,
  file = NULL,
  level = "reg",
  products = "kall",
  product_aggr = FALSE,
  attributes = "dm",
  water_aggr = TRUE,
  dir = ".",
  cumulative = FALSE,
  baseyear = 1995
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in gdxAggregate |
| `products` | Selection of products (either by naming products, e.g. "tece", or naming a set,e.g."kcr") |
| `product_aggr` | aggregate over products or not (boolean) |
| `attributes` | dry matter: Mt ("dm"), gross energy: PJ ("ge"), reactive nitrogen: Mt ("nr"), phosphor: Mt ("p"), potash: Mt ("k"), wet matter: Mt ("wm"). Can also be a vector. |
| `water_aggr` | aggregate irrigated and non-irriagted production or not (boolean). |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |
| `cumulative` | Logical; Determines if production is reported annually (FALSE, default) or cumulative (TRUE) |
| `baseyear` | Baseyear used for cumulative production (default = 1995) |

### Value

production as MAgPIE object (unit depends on attributes and cumulative)

### Author(s)

Benjamin Leon Bodirsky

### See Also

`reportProduction`, `demand`

### Examples

```
## Not run:
x <- production(gdx)

## End(Not run)
```

---

## `productionProfit` *productionProfit*

### Description

calcluates aggregate producer profit based on a MAgPIE gdx file.

### Usage

```
productionProfit(gdx, file = NULL, level = "reg", dir = ".")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |

### Value

A MAgPIE object containing producers profit in million $US.

### Author(s)

Miodrag Stevanovic

## Examples

```
  ## Not run:
    x <- productionProfit(gdx)

## End(Not run)
```

---

## productionRevenue *productionRevenue*

### Description

calcluates production revenue based on a MAgPIE gdx file.

### Usage

```
productionRevenue(
  gdx,
  file = NULL,
  level = "reg",
  products = "kall",
  product_aggr = TRUE
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `products` | Selection of products (either by naming products, e.g. "tece", or naming a set,e.g."kcr") |
| `product_aggr` | ggregate over products or not (boolean, default TRUE) |

### Value

A MAgPIE object containing prodcution revenues.

### Author(s)

Miodrag Stevanovic

### Examples

```
  ## Not run:
    x <- productionRevenue(gdx)

## End(Not run)
```

## `protectedArea` *protectedArea*

### Description

reads protectedArea out of a MAgPIE gdx file

### Usage

```
protectedArea(gdx, file = NULL, level = "cell", sum = FALSE, dir = ".")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "cell", "grid", "iso, "reg" (regional), "glo" (global), "regglo" (regional and global) or any secdforest aggregation level defined in superAggregate |
| `sum` | sum over land pools (default = FALSE) |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |

### Details

protected areas in primforest, secdforest and other land

### Value

protected area in Mha

### Author(s)

Florian Humpenoeder, Patrick v. Jeetze

### Examples

```
## Not run:
x <- protectedArea(gdx)

## End(Not run)
```

---

`relativeHourlyLaborCosts` *relativeHourlyLaborCosts*

---

## Description

calculates labor costs per ag. worker in relation to GDP pc

## Usage

```
relativeHourlyLaborCosts(gdx, level = "reg", file = NULL)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | spatial aggregation to report ("iso", "reg", "glo", or "regglo") |
| `file` | a file name the output should be written to using write.magpie |

## Value

labor costs per ag. worker in relation to GDP pc

## Author(s)

Debbora Leip

## Examples

```
## Not run:
x <- relativeHourlyLaborCosts(gdx)

## End(Not run)
```

---

`reportAAI` *reportAAI*

---

## Description

reports area actually irrigated

## Usage

```
reportAAI(gdx)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

Area actually irrigated as MAgPIE object. Unit: see names

## Author(s)

Stephen Wirth, Anne Biewald

## Examples

```
## Not run:
x <- reportAEI(gdx)

## End(Not run)
```

---

# `reportAEI` *reportAEI*

## Description

reports Area equipped for Irrigation

## Usage

```
reportAEI(gdx)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

Area equipped for Irrigation as MAgPIE object. Unit: see names

## Author(s)

Stephen Wirth

## Examples

```
  ## Not run:
    x <- reportAEI(gdx)

## End(Not run)
```

---

`reportAgEmployment` *reportAgEmployment*

---

## Description

reports employment in crop+livestock production from MAgPIE results

## Usage

```
reportAgEmployment(
  gdx,
  type = "absolute",
  detail = FALSE,
  level = "regglo",
  dir = "."
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `type` | "absolute" for total number of people employed, "share" for share out of working age population |
| `detail` | if TRUE, employment is disaggregated to crop and livestock production, if FALSE only aggregated employment is reported |
| `level` | spatial aggregation: "reg", "glo", "regglo", "iso" |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |

## Value

agricultural employment as MAgPIE object

## Author(s)

Debbora Leip

## Examples

```
  ## Not run:
    x <- reportAgEmployment(gdx)

## End(Not run)
```

---

## `reportAgGDP` *reportAgGDP*

### Description

reports MAgPIE Agricultural GDP Mio. USD05 MER

### Usage

```
reportAgGDP(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

Magpie object

### Author(s)

Edna J. Molina Bacca

### Examples

```
## Not run:
x <- reportAgGDP(gdx)

## End(Not run)
```

---

## `reportAgriResearchIntensity` *reportAgriResearchIntensity*

### Description

reports Agricultural Research Intensity as

### Usage

```
reportAgriResearchIntensity(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

magpie object

## Author(s)

David Chen

## Examples

```
## Not run:
x <- reportAgriResearchIntensity(gdx)

## End(Not run)
```

---

# `reportAnthropometrics` *reportAnthropometrics*

## Description

reports Underweight, Normalweight, Overweight and Obesity as well as body height for males and females

## Usage

```
reportAnthropometrics(gdx, level = "regglo")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | spatial aggregation: "reg", "glo", "regglo", "iso" |

## Value

Magpie object

## Author(s)

Benjamin Leon Bodirsky

## Examples

```
## Not run:
x <- reportBodyweight(gdx)

## End(Not run)
```

---

## `reportBII` *reportBII*

---

### Description

reports biodiversity intactness index

### Usage

```
reportBII(gdx, dir = ".")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `dir` | magpie output directory that contains gridded BII data |

### Value

Biodiversity intactness index as MAgPIE object

### Author(s)

Patrick v. Jeetze, Florian Humpenoeder

### Examples

```
## Not run:
x <- reportBII(gdx)

## End(Not run)
```

---

## `reportBioplasticDemand` *reportBioplasticDemand*

---

### Description

reports demand for bioplastic and demand for substrate for bioplastic production from MAgPIE results

### Usage

```
reportBioplasticDemand(gdx, detail = TRUE, level = "regglo")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `detail` | only relevant for substrate demand. If TRUE, substrate demand is disaggregated by crop type, if FALSE only the aggregated demand is reported. |
| `level` | spatial aggregation to report bioplastic/substrate demand (only "reg" or "regglo") |

## Value

bioplastic and bioplastic substrate demand as MAgPIE object

## Author(s)

Debbora Leip

## Examples

```
  ## Not run:
    x <- reportBioplasticDemand(gdx)

## End(Not run)
```

---

# `reportCarbonstock` *reportCarbonstock*

## Description

Reports the carbon stocks for future MAgPIE projections

## Usage

```
reportCarbonstock(gdx)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Author(s)

Kristine Karstens

## Examples

```
  ## Not run:
    x <- reportSOM(gdx)

## End(Not run)
```

---

## `reportConsumVal` *reportConsumVal*

---

### Description

reports MAgPIE consumption value

### Usage

```
reportConsumVal(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

Magpie object associated with the consumption value

### Author(s)

Edna J. Molina Bacca

### Examples

```
## Not run:
x <- reportConsumVal(gdx)

## End(Not run)
```

---

## `reportCostCapitalInvestment` *reportCostCapitalInvestment*

---

### Description

reports MAgPIE capital investments

### Usage

```
reportCostCapitalInvestment(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

Magpie object associated with overall costs and value of production

## Author(s)

Edna J. Molina Bacca

## Examples

```
## Not run:
x <- reportCostCapitalInvestment(gdx)

## End(Not run)
```

---

# `reportCostCapitalStocks` *reportCostCapitalStocks*

## Description

reports MAgPIE capital stocks

## Usage

```
reportCostCapitalStocks(gdx)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

Magpie object associated with overall costs and value of production

## Author(s)

Edna J. Molina Bacca

## Examples

```
## Not run:
x <- reportCostCapitalStocks(gdx)

## End(Not run)
```

---

## `reportCostInputsCrop` *reportCostInputsCrop*

### Description

reports MAgPIE costs

### Usage

```
reportCostInputsCrop(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

Magpie object associated with overall costs and value of production

### Author(s)

Edna J. Molina Bacca

### Examples

```
## Not run:
x <- reportCostInputsCrop(gdx)

## End(Not run)
```

---

## `reportCostOverall` *reportCostOverall*

### Description

reports MAgPIE costs

### Usage

```
reportCostOverall(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

Magpie object associated with overall costs and value of production

## Author(s)

Edna J. Molina Bacca

## Examples

```
  ## Not run:
    x <- reportCostOverall(gdx)

## End(Not run)
```

---

# `reportCosts` *reportCosts*

## Description

reports MAgPIE costs

## Usage

```
reportCosts(gdx)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

consumption value as MAgPIE object Unit: see names

## Author(s)

Florian Humpenoeder

## Examples

```
  ## Not run:
    x <- reportCosts(gdx)

## End(Not run)
```

---

## `reportCostsAccounting` *reportCostsAccounting*

---

### Description

reports MAgPIE costs including total investments

### Usage

```
reportCostsAccounting(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

Costs accounting including total investments

### Author(s)

Edna J. Molina Bacca

### Examples

```
## Not run:
x <- reportCostsAccounting(gdx)

## End(Not run)
```

---

## `reportCostsAEI` *reportCostsAEI*

---

### Description

reports MAgPIE AEI costs

### Usage

```
reportCostsAEI(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

magpie object containing AEI costs

### Author(s)

Felicitas Beier

### Examples

```
  ## Not run:
    x <- reportCostsAEI(gdx)

## End(Not run)
```

---

## `reportCostsFertilizer` *reportCostsFertilizer*

### Description

reports MAgPIE nitrogen fertilizer costs disaggregated to crop categories

### Usage

```
reportCostsFertilizer(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

magpie object with fertilizer costs

### Author(s)

Debbora Leip

### Examples

```
  ## Not run:
    x <- reportCostsFertilizer(gdx)

## End(Not run)
```

---

## `reportCostsInputFactors` *reportFactorCosts*

---

### Description

reports MAgPIE factor costs (split into labor and capital for sticky realization)

### Usage

```
reportCostsInputFactors(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

magpie object with factor costs

### Author(s)

Debbora Leip

### Examples

```
## Not run:
x <- reportCostsInputFactors(gdx)

## End(Not run)
```

---

## `reportCostsMACCS` *reportCostsMACCS*

---

### Description

reports MAgPIE mitigation costs disaggregated into labor and capital

### Usage

```
reportCostsMACCS(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

magpie object with mitigation costs

## Author(s)

Debbora Leip

## Examples

```
  ## Not run:
    x <- reportCostsMACCS(gdx)

## End(Not run)
```

---

# `reportCostsPresolve` *reportCostsPresolve*

## Description

reports MAgPIE costs

## Usage

```
reportCostsPresolve(gdx)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

consumption value as MAgPIE object Unit: see names

## Author(s)

Florian Humpenoeder

## Examples

```
  ## Not run:
    x <- reportCostsPresolve(gdx)

## End(Not run)
```

---

## `reportCostsProductionCrops` *reportCostsProductionCrops*

---

### Description

reports costs about production crops

### Usage

```
reportCostsProductionCrops(gdx, type = "investment")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `type` | Type of reporting, either "annuity" or total "investments" |

### Value

costs related to crops production, (million US$17/yr/tDM)

### Author(s)

Edna J. Molina Bacca

### Examples

```
## Not run:
x <- reportCostsProductionCrops(gdx)

## End(Not run)
```

---

## `reportCostsWholesale` *reportCostsWholesale*

---

### Description

Reads data to calculate wholesale costs

### Usage

```
reportCostsWholesale(gdx, level = "regglo")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |

## Value

A MAgPIE object containing values related with costs wholesale trade [million US$17/tDM]

## Author(s)

David M Chen

## Examples

```
## Not run:
x <- reportCostsWholesale(gdx)

## End(Not run)
```

---

`reportCostsWithoutIncentives`

# *reportCostsWithoutIncentives*

---

## Description

reports Costs Without Incentives

## Usage

```
reportCostsWithoutIncentives(gdx, level = "regglo")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | spatial aggregation: "reg", "glo", "regglo" |

## Value

magpie object

## Author(s)

David Chen

## Examples

```
## Not run:
x <- reportCostsWithoutIncentives(gdx)

## End(Not run)
```

---

# `reportCostTC` *reportCostTC*

## Description

reports MAgPIE TC costs

## Usage

```
reportCostTC(gdx)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

magpie object with TC costs

## Author(s)

David Chen

## Examples

```
## Not run:
x <- reportCostTC(gdx)

## End(Not run)
```

---

## `reportCostTransport` *reportCostTransport*

---

### Description

reports MAgPIE costs

### Usage

```
reportCostTransport(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

consumption value as MAgPIE object Unit: see names

### Author(s)

David Chen

### Examples

```
  ## Not run:
    x <- reportCostTransport(gdx)

## End(Not run)
```

---

## `reportCroparea` *reportCroparea*

---

### Description

reports croparea

### Usage

```
reportCroparea(gdx, detail = FALSE)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `detail` | if detail=FALSE, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops") |

## Value

Croparea as MAgPIE object (million Ha/yr)

## Author(s)

Florian Humpenoeder

## Examples

```
## Not run:
x <- reportCroparea(gdx)

## End(Not run)
```

---

# `reportCropareaGrid` *reportCropareaGrid*

## Description

reports croparea

## Usage

```
reportCropareaGrid(gdx, dir = ".", spamfiledirectory = "")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |
| `spamfiledirectory` | deprecated. please use `dir` instead |

## Value

Croparea as MAgPIE object (million ha/yr)

## Author(s)

Benjamin Bodirsky

## Examples

```
## Not run:
x <- reportCropareaGrid(gdx)

## End(Not run)
```

---

## `reportCropDiversity` *reportCropDiversity*

---

### Description

reports crop diversity

### Usage

```
reportCropDiversity(gdx, grid = FALSE, dir = ".")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `grid` | Set to TRUE, if outputs should be reported on 0.5 degree grid level |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |

### Value

Crop diversity as MAgPIE object

### Author(s)

Patrick v. Jeetze

### Examples

```
## Not run:
x <- reportCropDiversity(gdx)

## End(Not run)
```

---

## `reportDemand` *reportDemand*

---

### Description

reports Demand for Food, Feed, Processing, Material, Bioenergy, Seed and Supply Chain Loss

### Usage

```
reportDemand(gdx, detail = FALSE, agmip = FALSE)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `detail` | if detail=F, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops") |
| `agmip` | if agmip=T, additional sector aggregates required for agmip are reported (e.g. "AGR") |

### Value

demand as MAgPIE object (Mt DM)

### Author(s)

Benjamin Leon Bodirsky, Isabelle Weindl

### Examples

```
  ## Not run:
    x <- reportDemand()

## End(Not run)
```

---

## `reportDemandBioenergy` *reportDemandBioenergy*

### Description

reports Bioenergy Demand in EJ/yr

### Usage

```
reportDemandBioenergy(gdx, detail = FALSE)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `detail` | if detail=F, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops") |

### Value

Bioenergy demand as MAgPIE object (EJ/yr)

### Author(s)

Florian Humpenoeder, Kristine Karstens

## Examples

```
  ## Not run:
    x <- reportDemandBioenergy()

## End(Not run)
```

---

`reportDemandNr` *reportDemandNr*

## Description

Similar to reportDemand, but for nitrogen. reports Demand for Food, Feed, Processing, Material, Bioenergy, Seed and Supply Chain Loss

## Usage

```
reportDemandNr(gdx, detail = FALSE)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `detail` | if detail=F, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops") |

## Value

demand as MAgPIE object (Mt DM)

## Author(s)

Benjamin Leon Bodirsky

## Examples

```
  ## Not run:
    x <- reportDemand()

## End(Not run)
```

---

## `reportEmissions` *reportEmissions*

---

### Description

reports GHG emissions

### Usage

```
reportEmissions(gdx, storageWood = TRUE)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `storageWood` | Accounting for long term carbon storage in wood products. Default is TRUE. |

### Value

GHG emissions as MAgPIE object (Unit: Mt CO2/yr, Mt N2O/yr, and Mt CH4/yr, for cumulative emissions Gt CO2)

### Tier-1 variables

low pass filter = 3

| Name | Unit | Meta |
|---|---|---|
| Emissions\|CO2\|+\|Land | Mt CO2/yr | direct and indirect human-induced CO2 emissions from la |
| Emissions\|CO2\|Land\|+\|Indirect | Mt CO2/yr | indirect human-induced CO2 emissions from land use (la |
| Emissions\|CO2\|Land\|+\|Land-use Change | Mt CO2/yr | direct human-induced CO2 emissions from land use chan |
| Emissions\|CO2\|Land\|Land-use Change\|+\|Regrowth | Mt CO2/yr | negative CO2 emissions from regrowth |

### Tier-2 variables

raw data; no low pass filter applied

| Name | Unit | Meta |
|---|---|---|
| Emissions\|CO2\|+\|Land RAW | Mt CO2/yr | direct and indirect human-induced CO2 emissions from land u |
| Emissions\|CO2\|Land\|+\|Indirect RAW | Mt CO2/yr | indirect human-induced CO2 emissions from land use (land ca |
| Emissions\|CO2\|Land\|+\|Land-use Change RAW | Mt CO2/yr | direct human-induced CO2 emissions from land use change, h |

### Author(s)

Florian Humpenoeder, Benjamin Leon Bodirsky, Michael Crawford

## Examples

```
## Not run:
x <- reportEmissions(gdx)

## End(Not run)
```

---

`reportEmissionsBeforeTechnicalMitigation`

## *reportEmissionsBeforeTechnicalMitigation*

---

### Description

reports GHG emissions before technical mitigation. Technical abatement includes all abatement done in the MACC curves, but exclude endogenous mitigation. These emissions are NOT the standard reporting emissions, but used for special purposes like remind-magpie coupling.

### Usage

```
reportEmissionsBeforeTechnicalMitigation(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

MAgPIE object (Unit: Mt CO2/yr, Mt N2O/yr and Mt CH4/yr)

### Author(s)

Florian Humpenoeder, Benjamin Leon Bodirsky

### Examples

```
  ## Not run:
    x <- reportEmissionsBeforeTechnicalMitigation(gdx)

## End(Not run)
```

---

## `reportExpenditureFoodIndex` *reportExpenditureFoodIndex*

### Description

reports food expenditure index and food expenditure index corrected for emission costs

### Usage

```
reportExpenditureFoodIndex(gdx, baseyear = "y2010", basketyear = "y2010")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `baseyear` | Baseyear of the price index |
| `basketyear` | Year of reference food basket (should be in the past for comparison of different runs to have identical and comparable food basket) |

### Value

Food expenditure index as MAgPIE object

### Author(s)

Felicitas Beier

### Examples

```
## Not run:
x <- reportPriceFoodIndex(gdx)

## End(Not run)
```

---

## `reportFactorCostShares` *reportFactorCostShares*

### Description

reports labor and capital cost share out of factor costs from MAgPIE results

### Usage

```
reportFactorCostShares(gdx, type = "optimization", level = "regglo")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `type` | • "requirements": shares from factor requirements<br>• "optimization": cost shares between labor and capital costs in optimization<br>• "accounting": cost shares based on accounting of labor and capital costs |
| `level` | spatial aggregation: "reg", "glo", "regglo" |

## Value

labor and capital cost shares as MAgPIE object

## Author(s)

Debbora Leip

## Examples

```
## Not run:
x <- reportFactorCostShares(gdx)

## End(Not run)
```

---

# `reportFeed` *reportFeed*

## Description

reportes feed demand by animal type

## Usage

```
reportFeed(gdx, detail = T)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `detail` | if detail=F, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops") |

## Value

feed demand as MAgPIE object (Mt DM)

## Author(s)

Isabelle Weindl

## Examples

```
  ## Not run:
    x <- reportFeed()

## End(Not run)
```

---

`reportFeedConversion` *reportFeedConversion*

## Description

reportes feed demand by animal type

## Usage

```
reportFeedConversion(gdx, livestockSystem = TRUE, balanceflow = FALSE)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `livestockSystem` | if TRUE, ruminant products and poultry products are aggregated |
| `balanceflow` | If true, feed includes the calibration balanceflow |

## Value

feed demand as MAgPIE object (Mt DM)

## Author(s)

Benjamin Bodirsky

## Examples

```
  ## Not run:
    x <- reportFeed()

## End(Not run)
```

---

## `reportFertilizerNitrogen` *reportFertilizerNitrogen*

---

### Description

Reports inorganic nitrogen application on crops

### Usage

```
reportFertilizerNitrogen(gdx, level = "regglo")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | level of output |

### Author(s)

David M Chen

### See Also

`NitrogenBudget`

### Examples

```
## Not run:
x <- reportFertilizerNitrogen(gdx)

## End(Not run)
```

---

## `reportFit` *reportFit*

---

### Description

reports fit and error indicators compared to initial values

### Usage

```
reportFit(gdx, type = "MAPE", level = "cell")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `type` | type of indicator. Options: R2, MAE, MPE (mean percentage error - bias), MAPE (mean absolute percentage error) |
| `level` | level at which the regional and global bias should be reported. Options "cell" or "grid" |

## Value

Selected error indicator

## Author(s)

Edna Molina Bacca, Patrick v. Jeetze

## Examples

```
  ## Not run:
    x <- reportFit(gdx,type)

## End(Not run)
```

---

# `reportFoodExpenditure` *reportFoodExpenditure*

## Description

reports per-capita calories food supply (including household waste)

## Usage

```
reportFoodExpenditure(gdx, detail = FALSE, level = "regglo")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `detail` | if detail=F, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops") |
| `level` | spatial aggregation: "reg", "glo", "regglo", "iso" |

## Value

per-capita calories as MAgPIE object (kcal/cap/day)

## Author(s)

Benjamin Leon Bodirsky

## Examples

```
  ## Not run:
    x <- reportFoodExpenditure(gdx)

## End(Not run)
```

---

`reportForestYield` *reportForestYield*

---

## Description

reports MAgPIE harvested area for timber.

## Usage

```
reportForestYield(gdx)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

Yield from Forests for timber production

## Author(s)

Abhijeet Mishra

## Examples

```
  ## Not run:
    x <- reportForestYield(gdx)

## End(Not run)
```

---

`reportGraslandSoilCarbon`

## *reportGraslandSoilCarbon*

---

### Description

reports cattle related numbers

### Usage

```
reportGraslandSoilCarbon(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

Cattle values as magpie objetc

### Author(s)

Marcos Alves

### Examples

```
## Not run:
x <- reportGraslandSoilCarbon(gdx)

## End(Not run)
```

---

`reportGrasslandManagement`

## *reportGrasslandManagement*

---

### Description

reports cattle related numbers

### Usage

```
reportGrasslandManagement(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

Cattle values as magpie objetc

## Author(s)

Marcos Alves

## Examples

```
## Not run:
x <- reportGrasslandManagement(gdx)

## End(Not run)
```

---

# `reportGrasslandYields` *reportGrasslandYields*

## Description

reportGrasslandYields

## Usage

```
reportGrasslandYields(gdx)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

yield as MAgPIE object (Mt DM/ha)

## Author(s)

Marcos Alves

## Examples

```
  ## Not run:
    x <- reportGrasslandYields(gdx)

## End(Not run)
```

---

`reportGrassStats` *reportGrassStats*

---

## Description

report evaluation values for pasture management implementation

## Usage

```
reportGrassStats(gdx)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

magpie object

## Author(s)

Marcos Alves

## Examples

```
## Not run:
x <- reportGrassStats(gdx)

## End(Not run)
```

---

`reportGridCroparea` *reportGridCroparea*

---

## Description

reports Croparea from gridded (disaggregated) output

## Usage

```
reportGridCroparea(gdx, dir = ".", spamfiledirectory = "")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |
| `spamfiledirectory` | deprecated. please use `dir` instead |

## Value

area of cropland as MAgPIE object (million ha)

## Author(s)

Jannes Breier

## Examples

```
  ## Not run:
    x <- reportGridCroparea(gdx)

## End(Not run)
```

---

# `reportGridLand` *reportGridLand*

## Description

reports land-use from gridded (disaggregated) output

## Usage

```
reportGridLand(gdx, dir = ".", spamfiledirectory = "")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |
| `spamfiledirectory` | deprecated. please use `dir` instead |

## Value

land-use as MAgPIE object (million ha)

## Author(s)

Jannes Breier

## Examples

```
  ## Not run:
    x <- reportGridLand(gdx)

## End(Not run)
```

---

`reportGridManureExcretion`

# *reportGridManureExcretion*

---

## Description

reports Manure with reprting names on grid level.

## Usage

```
reportGridManureExcretion(gdx, dir = ".", spamfiledirectory = "")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |
| `spamfiledirectory` | deprecated. please use `dir` instead |

## Value

MAgPIE object

## Author(s)

Benjamin Leon Bodirsky

## Examples

```
## Not run:
x <- reportGridManureExcretion(gdx)

## End(Not run)
```

---

## `reportGrowingStock` *reportGrowingStock*

### Description

reports Growing stocks for woody materials

### Usage

```
reportGrowingStock(gdx, indicator = "relative", detail = FALSE)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `indicator` | If the reported numbers are relative (mio m3/ha) or absolute (mio. m3). Default is relative. |
| `detail` | if detail=FALSE, the subcategories of groups are not reported. |

### Value

production as MAgPIE object. Unit: see names

### Author(s)

Abhijeet Mishra

### Examples

```
  ## Not run:
    x <- reportGrowingStock(gdx)

## End(Not run)
```

---

## `reportharvested_area_timber` *reportharvested_area_timber*

### Description

reports MAgPIE harvested area for timber.

### Usage

```
reportharvested_area_timber(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

Area harvested for timber production

### Author(s)

Abhijeet Mishra

### Examples

```
  ## Not run:
    x <- reportharvested_area_timber(gdx)

## End(Not run)
```

---

## `reportHourlyLaborCosts` *reportHourlyLaborCosts*

### Description

reports hourly labor costs in agriculture from MAgPIE results

### Usage

```
reportHourlyLaborCosts(gdx, level = "regglo")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | spatial aggregation: "reg", "glo", "regglo" |

### Value

hourly labor costs as MAgPIE object

### Author(s)

Debbora Leip

## Examples

```
  ## Not run:
    x <- reportHourlyLaborCosts(gdx)

## End(Not run)
```

---

## reportHunger *reportHunger*

---

### Description

Calculates the share of people living in hunger.

### Usage

```
reportHunger(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

magpie object with hunger (mio people) or hunger share

### Author(s)

Benjamin Leon Bodirsky

### Examples

```
  ## Not run:
    x <- reportHunger(gdx)

## End(Not run)
```

---

## `reportIncome` *reportIncome*

---

### Description

reports income

### Usage

```
reportIncome(gdx, type = "ppp", level = "regglo")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `type` | ppp for purchase power parity, mer for market exchange rate |
| `level` | spatial aggregation: "reg", "glo", "regglo", "iso" |

### Value

Annual per capita and total income as MAgPIE object (US$2017 MER/cap/yr and million US$17 PPP/yr)

### Author(s)

Florian Humpenoeder, Isabelle Weindl, Felicitas Beier

### Examples

```
## Not run:
x <- reportIncome(gdx)

## End(Not run)
```

---

## `reportIntakeDetailed` *reportIntakeDetailed*

---

### Description

reports detailed or aggregated per-capita kcal intake including exogenous scenarios

### Usage

```
reportIntakeDetailed(gdx, detail = TRUE, level = "regglo")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `detail` | if detail=F, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops") |
| `level` | spatial aggregation: "reg", "glo", "regglo", "iso" |

## Value

per-capita calorie intake as MAgPIE object (kcal/cap/day)

## Author(s)

Isabelle Weindl

## Examples

```
  ## Not run:
    x <- reportIntakeDetailed(gdx)

## End(Not run)
```

---

`reportKcal` *reportKcal*

---

## Description

reports per-capita calories food supply (including household waste)

## Usage

```
reportKcal(gdx, detail = FALSE, level = "regglo")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `detail` | if detail=F, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops") |
| `level` | spatial aggregation: "reg", "glo", "regglo", "iso" |

## Value

per-capita calories as MAgPIE object (kcal/cap/day)

## Author(s)

Benjamin Leon Bodirsky, Kristine karstens, Abhijeet Mishra

## Examples

```
  ## Not run:
    x <- reportKcal(gdx)

## End(Not run)
```

---

# `reportLaborCostsEmpl` *reportLaborCostsEmpl*

---

## Description

reports MAgPIE labor costs that go into employment calculation

## Usage

```
reportLaborCostsEmpl(gdx)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

magpie object with labor costs

## Author(s)

Debbora Leip

## Examples

```
  ## Not run:
    x <- reportLaborCostsEmpl(gdx)

## End(Not run)
```

---

`reportLaborProductivity`

## *reportLaborProductivity*

---

### Description

reports labor productivity in crop production

### Usage

```
reportLaborProductivity(
  gdx,
  productAggr = TRUE,
  type = "physical",
  level = "regglo"
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `productAggr` | Aggregate over products or not (boolean) |
| `type` | type of labor productivity, so far only physical (kg DM / h) |
| `level` | spatial aggregation: "reg", "glo", "regglo", "iso" |

### Value

labor productivity as MAgPIE object

### Author(s)

Debbora Leip

### Examples

```
  ## Not run:
    x <- reportLaborProductivity(gdx)

## End(Not run)
```

---

## `reportLandUse` *reportLandUse*

---

### Description

reports land-use

### Usage

```
reportLandUse(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

land-use as MAgPIE object (million ha)

### Author(s)

Florian Humpenoeder, Kristine Karstens, Isabelle Weindl

### Examples

```
  ## Not run:
    x <- reportLandUse(gdx)

## End(Not run)
```

---

## `reportLandUseChange` *reportLandUseChange*

---

### Description

reports land-use change

### Usage

```
reportLandUseChange(gdx, baseyear = 1995)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `baseyear` | baseyear for calculating land-use change |

### Value

land-use change as MAgPIE object (million ha wrt to baseyear)

### Author(s)

Florian Humpenoeder

### Examples

```
  ## Not run:
    x <- reportLandUseChange(gdx)

## End(Not run)
```

---

## `reportLandUseForSEALS` *reportLandUseForSEALS*

### Description

Writes MAgPIE land use projections to a specific NetCDF that can be read by the Spatial Economic Allocation Landscape Simulator (SEALS) model for generating high resolution land use maps.

### Usage

```
reportLandUseForSEALS(
  magCellLand = "cell.land_0.5_share.mz",
  outFile = "cell.land_0.5_SEALS.nc",
  scenName = NULL,
  dir = ".",
  selectyears = c(2020, 2030, 2050)
)
```

### Arguments

| | |
|---|---|
| `magCellLand` | Disaggregated land use (grid-cell land area share) as magclass object or file (.mz) from a MAgPIE run. |
| `outFile` | a file name the output should be written to using `ncdf4::nc_create` and `ncdf4::ncvar_put` |
| `scenName` | Optional scenario name |
| `dir` | output directory which contains cellular magpie output |
| `selectyears` | Numeric vector of years to provide data for. |

### Value

Proportions of different land use classes per grid sell in a NetCDF format.

## Author(s)

Patrick v. Jeetze

## Examples

```
## Not run:
x <- reportLandUseForSEALS(
  magCellLand = "cell.land_0.5_share.mz",
  outFile = "cell.land_0.5_SEALS.nc",
  selectyears = c(2020, 2030, 2050)
)

## End(Not run)
```

---

`reportLivestockDemStructure`

# *reportLivestockDemStructure*

---

## Description

reports the share of different livestock products (excluding fish) in total livestock calorie food supply

## Usage

```
reportLivestockDemStructure(gdx)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

livestock demand structure as MAgPIE object (kcal/kcal)

## Author(s)

Isabelle Weindl

## Examples

```
  ## Not run:
    x <- reportLivestockDemStructure(gdx)

## End(Not run)
```

---

## `reportLivestockShare` *reportLivestockShare*

### Description

reports the share of livestock products (including fish) in total calorie food supply

### Usage

```
reportLivestockShare(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

per-capita calories as MAgPIE object (kcal/cap/day)

### Author(s)

Benjamin Leon Bodirsky

### Examples

```
  ## Not run:
    x <- reportLivestockShare(gdx)

## End(Not run)
```

---

## `reportLSUGrasslands` *reportLSUGrasslands*

### Description

reportLSUGrasslands

### Usage

```
reportLSUGrasslands(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

Livestock eq. denstity as a magpie object (Mt DM/ha) (1 LSU eq. = 8.9 kg DM/day)

## Author(s)

Marcos Alves

## Examples

```
  ## Not run:
    x <- reportGrasslandYields(gdx)

## End(Not run)
```

---

# reportManure

`reportManure` *reportManure*

## Description

Reports the Nitrogen in Manure of all animals for future MAgPIE projections

Reports the Nitrogen in Manure of all animals for future MAgPIE projections

## Usage

```
reportManure(gdx, nutrient = "nr")

reportManure(gdx, nutrient = "nr")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `nutrient` | nr, p, c... |

## Author(s)

Benjamin Leon Bodirsky

## Examples

```
  ## Not run:
    x <- reportManure(gdx)

## End(Not run)

  ## Not run:
    x <- reportManure(gdx)

## End(Not run)
```

---

## `reportNetForestChange` *reportNetForestChange*

---

### Description

reports Net Forest Change

### Usage

```
reportNetForestChange(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

NetForestChange as magclass object (Mha per year)

### Author(s)

Florian Humpenoeder

### Examples

```
## Not run:
x <- reportNetForestChange(gdx)

## End(Not run)
```

---

## `reportNitrogenBudgetCropland` *reportNitrogenBudgetCropland*

---

### Description

Reports the Nitrogen Budgets of Croplands for future MAgPIE projections

### Usage

```
reportNitrogenBudgetCropland(
  gdx,
  include_emissions = FALSE,
  grid = FALSE,
  dir = "."
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `include_emissions` | TRUE also divides the N surplus into different emissions |
| `grid` | grid provides outputs on grid level of 0.5 degree |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |

## Author(s)

Benjamin Leon Bodirsky

## See Also

`NitrogenBudget`

## Examples

```
## Not run:
x <- reportNitrogenBudgetCropland(gdx)

## End(Not run)
```

---

`reportNitrogenBudgetNonagland`

# *reportNitrogenBudgetNonagland*

## Description

Reports the Nitrogen Budgets of non-agricultural lands for future MAgPIE projections

## Usage

```
reportNitrogenBudgetNonagland(gdx, grid = FALSE, dir = ".")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `grid` | if TRUE, disaggregate to grid level |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |

## Author(s)

Benjamin Leon Bodirsky

## See Also

NitrogenBudget

## Examples

```
  ## Not run:
    x <- reportNitrogenBudgetNonagland(gdx)

## End(Not run)
```

---

# `reportNitrogenBudgetPasture` *reportNitrogenBudgetCropland*

## Description

Reports the Nitrogen Budgets of Croplands for future MAgPIE projections

## Usage

```
reportNitrogenBudgetPasture(
  gdx,
  include_emissions = FALSE,
  grid = FALSE,
  dir = "."
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `include_emissions` | TRUE also divides the N surplus into different emissions |
| `grid` | if TRUE, disaggregate to grid level |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |

## Author(s)

Benjamin Leon Bodirsky

## See Also

NitrogenBudget

## Examples

```
  ## Not run:
    x <- reportNitrogenBudgetCropland(gdx)

## End(Not run)
```

---

`reportNitrogenEfficiencies`

# *reportNitrogenEfficiencies*

---

## Description

Reports different nitrogen use efficiency indicators

## Usage

```
reportNitrogenEfficiencies(gdx)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Author(s)

Benjamin Leon Bodirsky

## See Also

`reportNitrogenEfficiencies`

## Examples

```
## Not run:
x <- reportNitrogenEfficiencies(gdx)

## End(Not run)
```

---

## `reportNitrogenPollution` *reportNitrogenPollution*

---

### Description

Reports total Nitrogen Pollution as the sum of surplus from cropland, pasture, awms, consumption and non-agricutlural land

### Usage

```
reportNitrogenPollution(gdx, dir = ".")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `dir` | magpie output directory that contains gridded Nitrogen Data |

### Author(s)

Benjamin Leon Bodirsky

### See Also

`NitrogenBudget`

### Examples

```
## Not run:
x <- reportNitrogenPollution(gdx)

## End(Not run)
```

---

## `reportOutputPerWorker` *reportOutputPerWorker*

---

### Description

reports output per worker in crop+livestock production from MAgPIE results

### Usage

```
reportOutputPerWorker(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

output per worker as MAgPIE object

### Author(s)

Debbora Leip

### Examples

```
  ## Not run:
    x <- reportOutputPerWorker(gdx)

## End(Not run)
```

---

## `reportPastSoilCarbon` *reportPastSoilCarbon*

### Description

reports pasture soil carbon

### Usage

```
reportPastSoilCarbon(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

Soil carbon values as magpie objetc

### Author(s)

Marcos Alves

### Examples

```
## Not run:
x <- reportPastSoilCarbon(gdx)

## End(Not run)
```

---

`reportPBbiosphere` *reportPBbiosphere*

---

## Description

reports biosphere planetary boundary: Share of intact land relative to total land area (unitless) Share of intact land covered by areas within Global Safety Net (unitless) Share of land area that satisfies landscape target (unitless)

## Usage

```
reportPBbiosphere(
  gdx,
  level = "regglo",
  dir = ".",
  intactnessRule = "carbon:0.95"
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | level of aggregation (regglo: regions and global) |
| `dir` | directory with required spatial data |
| `intactnessRule` | rule for intact land can be based on percentage of potential carbon density reached or on age classes for secondary forests, planted forest and other natural land. The argument is split into two components: rule: carbon or ageclass threshold: share of carbon density reached to be classified as intact or threshold in years can be set via this argument Example: "carbon:0.95" or "ageclass:70" |

## Value

MAgPIE object

## Author(s)

Patrick von Jeetze, Felicitas Beier

## Examples

```
## Not run:
x <- reportPBbiosphere(gdx)

## End(Not run)
```

---

## `reportPBland` *reportPBland*

---

### Description

reports land planetary boundary: forest area as percentage of original forest area

### Usage

```
reportPBland(gdx, level = "regglo", dir = ".", foresttype = "all")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | level of aggregation (regglo: regions and global) |
| `dir` | directory with required spatial data |
| `foresttype` | managed forest types that are included in the calculation of the forest area (all: all managed forests, noTimber: timber plantations are not counted) |

### Value

MAgPIE object

### Author(s)

Felicitas Beier, Patrick von Jeetze

### Examples

```
  ## Not run:
    x <- reportPBland(gdx)

## End(Not run)
```

---

## `reportPBnitrogen` *reportPBnitrogen*

---

### Description

reports nitrogen planetary boundary

### Usage

```
reportPBnitrogen(gdx, level = "regglo", dir = ".")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | level of aggregation (regglo: regions and global) |
| `dir` | directory with required spatial data |

## Value

MAgPIE object

## Author(s)

Felicitas Beier, Mike Crawford

## Examples

```
  ## Not run:
    x <- reportPBnitrogen(gdx)

## End(Not run)
```

---

# `reportPBwater` *reportPBwater*

## Description

reports water planetary boundaries

## Usage

```
reportPBwater(gdx, level = "regglo")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | level of aggregation (regglo: regions and global) |

## Value

MAgPIE object

## Author(s)

Felicitas Beier, Jens Heinke

## Examples

```
  ## Not run:
    x <- reportPBwater(gdx)

## End(Not run)
```

---

`reportPeatland` *reportPeatland*

## Description

reports peatland area

## Usage

```
reportPeatland(gdx)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

peatland area as magclass object (million ha)

## Author(s)

Florian Humpenoeder

## Examples

```
  ## Not run:
    x <- reportPeatland(gdx)

## End(Not run)
```

---

`reportPlantationEstablishment` *reportPlantationEstablishment*

---

## Description

reports MAgPIE harvested area for timber.

## Usage

```
reportPlantationEstablishment(gdx)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

Area harvested for timber production

## Author(s)

Abhijeet Mishra

## Examples

```
  ## Not run:
    x <- reportPlantationEstablishment(gdx)

## End(Not run)
```

---

`reportPopulation` *reportPopulation*

---

## Description

reports Population

## Usage

```
reportPopulation(gdx, level = "regglo")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | spatial aggregation: "reg", "glo", "regglo", "iso" |

## Value

population as MAgPIE object

## Author(s)

Florian Humpenoeder

## Examples

```
  ## Not run:
    x <- reportPopulation(gdx)

## End(Not run)
```

---

`reportPriceAgriculture` *reportPriceAgriculture*

---

## Description

reports food commodity prices

## Usage

```
reportPriceAgriculture(gdx)
```

## Arguments

| | |
|---|---|
| gdx | GDX file |

## Value

agricultural commodity prices as MAgPIE object (USD)

## Author(s)

Mishko Stevanovic

## Examples

```
  ## Not run:
    x <- reportPriceAgriculture(gdx)

## End(Not run)
```

---

## `reportPriceBioenergy` *reportPriceBioenergy*

---

### Description

reports bioenergy prices

### Usage

```
reportPriceBioenergy(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

bioenergy price as MAgPIE object Unit: see names

### Author(s)

Florian Humpenoeder

### Examples

```
  ## Not run:
    x <- reportPriceBioenergy(gdx)

## End(Not run)
```

---

## `reportPriceElasticities` *reportHunger*

---

### Description

Calculates the share of people living in hunger.

### Usage

```
reportPriceElasticities(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

magpie object with hunger (mio people) or hunger share

## Author(s)

Benjamin Leon Bodirsky

## Examples

```
  ## Not run:
    x <- reportHunger(gdx)

## End(Not run)
```

---

# `reportPriceFoodIndex` *reportPriceFoodIndex*

## Description

reports food price index

## Usage

```
reportPriceFoodIndex(gdx, baseyear = "y2020")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `baseyear` | baseyear of the price index |

## Value

Food price index as MAgPIE object Unit: see names

## Author(s)

Florian Humpenoeder, Felicitas Beier

## Examples

```
  ## Not run:
    x <- reportPriceFoodIndex(gdx)

## End(Not run)
```

---

## `reportPriceGHG` *reportPriceGHG*

---

### Description

reports GHG emission prices

### Usage

```
reportPriceGHG(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

GHG emission prices as MAgPIE object

### Author(s)

Florian Humpenoeder, Amsalu W. Yalew

### Examples

```
  ## Not run:
    x <- reportPriceGHG(gdx)

## End(Not run)
```

---

## `reportPriceLand` *reportPriceLand*

---

### Description

reports land prices (land rent)

### Usage

```
reportPriceLand(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

land prices as MAgPIE object Unit: see names

## Author(s)

Florian Humpenoeder

## Examples

```
  ## Not run:
    x <- reportPriceLand(gdx)

## End(Not run)
```

---

# `reportPriceShock` *reportPriceShock*

## Description

Reports the change in consumption and expenditure due to higher or lower food prices

## Usage

```
reportPriceShock(gdx)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

magpie object

## Author(s)

Benjamin Leon Bodirsky

## Examples

```
  ## Not run:
    x <- reportPriceShock(gdx)

## End(Not run)
```

---

`reportPriceWater` *reportPriceWater*

---

## Description

reports water prices

## Usage

```
reportPriceWater(gdx)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

water usage as MAgPIE object Unit: see names

## Author(s)

Florian Humpenoeder

## Examples

```
## Not run:
x <- reportPriceWater(gdx)

## End(Not run)
```

---

`reportPriceWoodyBiomass` *reportPriceWoodyBiomass*

---

## Description

reports woody biomass prices (land rent)

## Usage

```
reportPriceWoodyBiomass(gdx)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

land prices as MAgPIE object Unit: see names

## Author(s)

Abhijeet Mishra

## Examples

```
  ## Not run:
    x <- reportPriceWoodyBiomass(gdx)

## End(Not run)
```

---

## `reportProcessing` *reportProcessing*

## Description

reportes processing input and output quantities primary-to-process or primary-to-secondary

## Usage

```
reportProcessing(gdx, detail = TRUE, indicator = "primary_to_process")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `detail` | if detail=FALSE, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops") |
| `indicator` | "primary_to_process" for process or "secondary_from_primary" for secondary product output |

## Value

processing demand as MAgPIE object (Mt DM)

## Author(s)

David Chen, Benjamin Leon Bodirsky

## Examples

```
## Not run:
x <- reportProcessing(gdx = gdx, detail = TRUE, indicator = "primary_to_process")

## End(Not run)
```

---

## `reportProducerPriceIndex` *reportProducerPriceIndex*

---

### Description

reports producer price index

### Usage

```
reportProducerPriceIndex(gdx, prod_groups = FALSE)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `prod_groups` | whether to return only product groups |

### Value

Producer price index as MAgPIE object Unit: see names

### Author(s)

Isabelle Weindl, David M CHen

### Examples

```
  ## Not run:
    x <- reportProducerPriceIndex(gdx)

## End(Not run)
```

---

## `reportProduction` *reportProduction*

---

### Description

reports production

### Usage

```
reportProduction(gdx, detail = FALSE, agmip = FALSE)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `detail` | if detail=FALSE, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops") |
| `agmip` | if agmip = TRUE, additional sector aggregates required for AgMIP are reported (e.g. "AGR") |

## Value

production as MAgPIE object. Unit: see names

## Author(s)

Benjamin Leon Bodirsky, Isabelle Weindl

## Examples

```
  ## Not run:
    x <- reportProduction(gdx)

## End(Not run)
```

---

`reportProductionBioenergy`

# *reportProductionBioenergy*

## Description

reports 2nd gen bioenergy production

## Usage

```
reportProductionBioenergy(gdx, detail = FALSE)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `detail` | if detail=FALSE, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops") |

## Value

production as MAgPIE object. Unit: see names

## Author(s)

Florian Humpenoeder

## Examples

```
  ## Not run:
    x <- reportProductionBioenergy(gdx)

## End(Not run)
```

---

`reportProductionGrowth` *reportProductionGrowth*

---

## Description

reports production growth rate

## Usage

```
reportProductionGrowth(gdx, detail = FALSE)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `detail` | if true, provides results for all commodities, otherwhise aggregates some groups |

## Value

Production growth rates (index)

## Author(s)

Xiaoxi Wang

## Examples

```
  ## Not run:
    x <- reportProductionGrowth(gdx="fulldata.gdx",detail=TRUE)

## End(Not run)
```

## `reportProductionNr` *reportProductionNr*

### Description

reports production in Nr analogous to reportProduction

### Usage

```
reportProductionNr(gdx, detail = FALSE)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `detail` | if detail = FALSE, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops") |

### Value

production as MAgPIE object. Unit: see names

### Author(s)

Benjamin Leon Bodirsky

### Examples

```
  ## Not run:
    x <- reportProductionNr(gdx)

## End(Not run)
```

## `reportProtectedArea` *reportProtectedArea*

### Description

reports protected areas

### Usage

```
reportProtectedArea(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

protected area in Mha

## Author(s)

Florian Humpenoeder

## Examples

```
## Not run:
x <- reportProtectedArea(gdx)

## End(Not run)
```

---

# `reportProtein` *reportProtein*

## Description

reports per-capita protein food supply (including household waste)

## Usage

```
reportProtein(gdx, detail = FALSE)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `detail` | if detail=F, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops") |

## Value

per-capita protein as MAgPIE object (protein/cap/day)

## Author(s)

Benjamin Leon Bodirsky, Kristine Karstens, Abhijeet Mishra, Florian Humpenoeder

## Examples

```
  ## Not run:
    x <- reportKcal(gdx)

## End(Not run)
```

---

## `reportRelativeHourlyLaborCosts` *reportRelativeHourlyLaborCosts*

---

### Description

reports labor costs per ag. worker in relation to GDP pc from MAgPIE results

### Usage

```
reportRelativeHourlyLaborCosts(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

labor costs per ag. worker in relation to GDP pc as MAgPIE object

### Author(s)

Debbora Leip

### Examples

```
  ## Not run:
    x <- reportRelativeHourlyLaborCosts(gdx)

## End(Not run)
```

---

## `reportResidues` *reportSOM*

---

### Description

Report soil organic carbon stock size for future MAgPIE projections

### Usage

```
reportResidues(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Author(s)

Kristine Karstens

## Examples

```
  ## Not run:
    x <- reportSOM(gdx)

## End(Not run)
```

---

# `reportRotationLength` *reportRotationLength*

## Description

reports Forest rotation length.

## Usage

```
reportRotationLength(gdx)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

Forest rotation length

## Author(s)

Abhijeet Mishra

## Examples

```
  ## Not run:
    x <- reportRotationLength(gdx)

## End(Not run)
```

---

`reportRuralDemandShares` *reportRuralDemandShares*

---

## Description

reports rural demand and production shares based on local consumption

## Usage

```
reportRuralDemandShares(gdx, type = "tradOnly", level = "regglo")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `type` | Type of ratio that should be calculated<br>• `all`: How much rural & trad demand as a share of all demand is satisfied locally<br>• `tradOnly`: How much rural & trad demand as a share of rural & trad demand is satisfied locally<br>• `potential`: How much total gridded demand is potentially satisfied by gridded production |
| `level` | spatial aggregation: "reg", "glo", "regglo" |

## Value

share of food demand at disaggregated level coming from local production as MAgPIE object

## Author(s)

David M Chen

## Examples

```
## Not run:
x <- reportruralDemandShares(gdx)

## End(Not run)
```

---

## `reportSDG1` *reportSDG1*

---

### Description

reports all SDG indicators relevant for SDG1 - Poverty

### Usage

```
reportSDG1(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

MAgPIE object

### Author(s)

Benjamin Bodirsky

### Examples

```
  ## Not run:
    x <- reportSDG3(gdx)

## End(Not run)
```

---

## `reportSDG12` *reportSDG12*

---

### Description

reports all SDG indicators relevant for SD12 - Sustainable Production and Consumption

### Usage

```
reportSDG12(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

MAgPIE object

## Author(s)

Benjamin Bodirsky

## Examples

```
  ## Not run:
    x <- reportSDG12(gdx)

## End(Not run)
```

---

# `reportSDG15` *reportSDG15*

## Description

reports all SDG indicators relevant for SD15 - Life on Land

## Usage

```
reportSDG15(gdx)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

MAgPIE object

## Author(s)

Benjamin Bodirsky, Isabelle Weindl

## Examples

```
  ## Not run:
    x <- reportSDG15(gdx)

## End(Not run)
```

---

## `reportSDG2` *reportSDG2*

---

### Description

reports all SDG indicators relevant for SD2 - Hunger

### Usage

```
reportSDG2(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

MAgPIE object

### Author(s)

Benjamin Bodirsky

### Examples

```
  ## Not run:
    x <- reportSDG2(gdx)

## End(Not run)
```

---

## `reportSDG3` *reportSDG3*

---

### Description

reports all SDG indicators relevant for SDG3 - Health

### Usage

```
reportSDG3(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

MAgPIE object

### Author(s)

Benjamin Bodirsky

### Examples

```
  ## Not run:
    x <- reportSDG3(gdx)

## End(Not run)
```

---

## `reportSDG6` *reportSDG6*

### Description

reports all SDG indicators relevant for SDG6 - Access to Water

### Usage

```
reportSDG6(gdx, level = "regglo", outputdir = ".")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | level of aggregation (cluster: "cell", regional: "regglo") |
| `outputdir` | output directory |

### Value

MAgPIE object

### Author(s)

Felicitas Beier, Isabelle Weindl

### Examples

```
  ## Not run:
    x <- reportSDG6(gdx)

## End(Not run)
```

---

## `reportSDG9` *reportSDG9*

### Description

reports all SDG indicators relevant for SD9 - Industrial innovation and infrastructure

### Usage

```
reportSDG9(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

MAgPIE object

### Author(s)

Felicitas Beier

### Examples

```
  ## Not run:
    x <- reportSDG9(gdx)

## End(Not run)
```

---

## `reportSOM` *reportSOM*

### Description

Report soil organic carbon stock size for future MAgPIE projections

### Usage

```
reportSOM(gdx, baseyear = 1995)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `baseyear` | baseyear for calculating carbon stock change |

### Author(s)

Kristine Karstens

### Examples

```
  ## Not run:
    x <- reportSOM(gdx)

## End(Not run)
```

---

## `reportSOM2` *reportSOM2*

### Description

Report soil organic carbon stock size for future MAgPIE projections (new som realization)

### Usage

```
reportSOM2(gdx, baseyear = 1995)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `baseyear` | baseyear for calculating carbon stock change |

### Author(s)

Kristine Karstens

### Examples

```
  ## Not run:
    x <- reportSOM2(gdx)

## End(Not run)
```

---

## `reportTau` *reportTau*

---

### Description

reports Tau

### Usage

```
reportTau(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

tau values as MAgPIE object (Index)

### Author(s)

Florian Humpenoeder

### Examples

```
## Not run:
x <- reportTau(gdx)

## End(Not run)
```

---

## `reportTc` *reportTc*

---

### Description

reports Tc

### Usage

```
reportTc(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

tc values as MAgPIE object (

### Author(s)

Florian Humpenoeder

### Examples

```
  ## Not run:
    x <- reportTc(gdx)

## End(Not run)
```

---

## `reportTimber` *reportTimber*

### Description

reports MAgPIE demand for timber.

### Usage

```
reportTimber(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

Timber demand

### Author(s)

Abhijeet Mishra

### Examples

```
  ## Not run:
    x <- reportTimber(gdx)

## End(Not run)
```

---

## `reportTimberDemand` *reportTimberDemand*

---

### Description

reports MAgPIE demand for timber.

### Usage

```
reportTimberDemand(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

Timber demand

### Author(s)

Abhijeet Mishra

### Examples

```
  ## Not run:
    x <- reportTimberDemand(gdx)

## End(Not run)
```

---

## `reportTimberDemandVolumetric` *reportTimberDemandVolumetric*

---

### Description

reports MAgPIE demand for timber.

### Usage

```
reportTimberDemandVolumetric(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

Timber demand

## Author(s)

Abhijeet Mishra

## Examples

```
## Not run:
  x <- reportTimberDemandVolumetric(gdx)

## End(Not run)
```

---

# reportTimberProductionVolumetric

*reportTimberProductionVolumetric*

## Description

reports MAgPIE production for timber.

## Usage

```
reportTimberProductionVolumetric(gdx)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

Timber demand

## Author(s)

Abhijeet Mishra

## Examples

```
## Not run:
  x <- reportTimberProductionVolumetric(gdx)

## End(Not run)
```

---

`reportTotalHoursWorked` *reportTotalHoursWorked*

---

## Description

reports total hours worked in crop+livestock production (and maccs) from MAgPIE results

## Usage

```
reportTotalHoursWorked(gdx, level = "regglo")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | spatial aggregation: "reg", "glo", "regglo" |

## Value

total hours worked as MAgPIE object

## Author(s)

Debbora Leip

## Examples

```
  ## Not run:
    x <- reportTotalHoursWorked(gdx)

## End(Not run)
```

---

`reportTrade` *reportTrade*

---

## Description

reports trade

## Usage

```
reportTrade(gdx, detail = FALSE)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `detail` | if true, provides estimates for all commodities, otherwise aggregates some groups |

### Value

Net-Exports and self sufficiency (exports/domestic supply) as MAgPIE object. Unit: see names

### Author(s)

Benjamin Leon Bodirsky, Mishko Stevanovic

### Examples

```
  ## Not run:
    x <- reportTrade(gdx="fulldata.gdx",detail=TRUE)

## End(Not run)
```

---

## `reportTradeGrowth` *reportTradeGrowth*

### Description

reports trade growth rate

### Usage

```
reportTradeGrowth(gdx, detail = FALSE)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `detail` | if true, provides results for all commodities, otherwhise aggregates some groups |

### Value

Trade growth rates (index)

### Author(s)

Xiaoxi Wang

### Examples

```
  ## Not run:
    x <- reportTradeGrowth(gdx="fulldata.gdx",detail=TRUE)

## End(Not run)
```

---

## `reportValueMaterialDemand` *reportValueMaterialDemand*

---

### Description

reports value of material demand

### Usage

```
reportValueMaterialDemand(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

magpie object

### Author(s)

David Chen

### Examples

```
## Not run:
x <- reportValueMaterialDemand(gdx)

## End(Not run)
```

---

## `reportValueTrade` *reportValueTrade*

---

### Description

reports trade value

### Usage

```
reportValueTrade(gdx, detail = FALSE)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `detail` | if true, provides estimates for all commodities, otherwhise aggregates some groups |

## Value

trade value as MAgPIE object Unit: see names

## Author(s)

Florian Humpenoeder

## Examples

```
  ## Not run:
    x <- reportValueTrade(gdx)

## End(Not run)
```

---

# `reportVegfruitShare` *reportVegfruitShare*

## Description

reports the share of livestock products (including fish) in total calorie food supply

## Usage

```
reportVegfruitShare(gdx)
```

## Arguments

| | |
|---|---|
| gdx | GDX file |

## Value

per-capita calories as MAgPIE object (kcal/cap/day)

## Author(s)

Benjamin Leon Bodirsky

## Examples

```
  ## Not run:
    x <- reportLivestockShare(gdx)

## End(Not run)
```

---

## `reportWageDevelopment` *reportWageDevelopment*

---

### Description

reports indicator on wage development: hourly labor costs in each time step relative to hourly labor costs in 2000

### Usage

```
reportWageDevelopment(gdx, baseYear = 2000, level = "regglo")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `baseYear` | year relative to which the wage development should be calculated |
| `level` | spatial aggregation: "reg", "glo", "regglo" |

### Value

indicator on wage development as MAgPIE object

### Author(s)

Debbora Leip

### Examples

```
## Not run:
x <- reportWageDevelopment(gdx)

## End(Not run)
```

---

## `reportWaterAvailability` *reportWaterAvailability*

---

### Description

reports water availability

### Usage

```
reportWaterAvailability(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

water availability as MAgPIE object Unit: see names

### Author(s)

Felicitas Beier

### Examples

```
  ## Not run:
    x <- reportWaterAvailability(gdx)

## End(Not run)
```

---

## `reportWaterIndicators` *reportWaterIndicators*

### Description

reports a set of water indicators

### Usage

```
reportWaterIndicators(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |

### Value

MAgPIE object

### Author(s)

Felicitas Beier

### Examples

```
## Not run:
x <- reportWaterIndicators(gdx)

## End(Not run)
```

---

## `reportWaterUsage` *reportWaterUsage*

---

### Description

reports water usage for agricultural sector, crops and livestock and non-agricultural sector

### Usage

```
reportWaterUsage(gdx, detail = TRUE)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `detail` | logical. Setting to FALSE reports for agricultural sector, TRUE reports for combined, crops and livestock separately |

### Value

water usage as MAgPIE object Unit: see names

### Author(s)

Florian Humpenoeder, Vartika Singh, Miodrag Stevanovic, Felicitas Beier

### Examples

```
## Not run:
x <- reportWaterUsage(gdx)

## End(Not run)
```

---

## `reportWorkingAgePopulation` *reportWorkingAgePopulation*

---

### Description

reports working age population

### Usage

```
reportWorkingAgePopulation(gdx, level = "regglo")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | spatial aggregation: "reg", "glo", "regglo", "iso" |

### Value

working age population as MAgPIE object

### Author(s)

Debbora Leip

### Examples

```
  ## Not run:
    x <- reportWorkingAgePopulation(gdx)

## End(Not run)
```

---

## `reportYields` *reportYields*

### Description

reports yields

### Usage

```
reportYields(gdx, detail = FALSE, physical = TRUE)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `detail` | if detail=FALSE, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops") |
| `physical` | if true (default) physical area (croparea) used for yield calculation; if false harvested area used for yield calculation |

### Value

yield as MAgPIE object (Mt DM/ha)

### Author(s)

Florian Humpenoeder, Xiaoxi Wang, Kristine Karstens, Abhijeet Mishra, Felicitas Beier

## Examples

```
## Not run:
x <- reportYields(gdx)

## End(Not run)
```

---

# `reportYieldsCropCalib` *reportYieldsCropCalib*

## Description

reports potential yields after calibration

## Usage

```
reportYieldsCropCalib(gdx, detail = FALSE)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `detail` | if detail=FALSE, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops") |

## Value

yield as MAgPIE object (Mt DM/ha)

## Author(s)

Edna J. Molina Bacca

## Examples

```
## Not run:
x <- reportYieldsCropCalib(gdx)

## End(Not run)
```

---

## `reportYieldsCropRaw` *reportYieldsCropRaw*

---

### Description

reports potential yields before calibration

### Usage

```
reportYieldsCropRaw(gdx, detail = FALSE)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `detail` | if detail=FALSE, the subcategories of groups are not reported (e.g. "soybean" within "oilcrops") |

### Value

yield as MAgPIE object (Mt DM/ha)

### Author(s)

Edna J. Molina Bacca

### Examples

```
## Not run:
x <- reportYieldsCropRaw(gdx)

## End(Not run)
```

---

## `reportYieldShifter` *reportYieldShifter*

---

### Description

Reports the Crop model input yield shifter

### Usage

```
reportYieldShifter(
  gdx,
  file = NULL,
  level = "reg",
  baseyear = "y2000",
  relative = TRUE,
  dir = ".",
  spamfiledirectory = ""
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `baseyear` | baseyear for the yield shifter. Also fixes land patterns for aggregation to baseyear. |
| `relative` | relative or absolute changes to baseyear |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |
| `spamfiledirectory` | deprecated. please use `dir` instead |

### Value

crop yield as MAgPIE object (unit depends on attributes)

### Author(s)

Benjamin Leon Bodirsky

### See Also

`reportYieldShifter`

### Examples

```
  ## Not run:
    x <- reportYieldShifter(gdx)

## End(Not run)
```

---

`ResidueBiomass` *ResidueBiomass*

---

## Description

reads Crop Residue Biomass out of a MAgPIE gdx file

## Usage

```
ResidueBiomass(
  gdx,
  level = "reg",
  dir = ".",
  spamfiledirectory = "",
  products = "kcr",
  product_aggr = FALSE,
  attributes = "dm",
  water_aggr = TRUE,
  plantpart = "both"
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |
| `spamfiledirectory` | deprecated. please use `dir` instead |
| `products` | Selection of products (either by naming products, e.g. "tece", or naming a set,e.g."kcr") |
| `product_aggr` | aggregate over products or not. Usually boolean, but here also the value "kres" is allowed, which provides kcr aggregated to kres |
| `attributes` | dry matter: Mt ("dm"), gross energy: PJ ("ge"), reactive nitrogen: Mt ("nr"), phosphor: Mt ("p"), potash: Mt ("k"), wet matter: Mt ("wm"). Can also be a vector. |
| `water_aggr` | aggregate irrigated and non-irriagted production or not (boolean). |
| `plantpart` | both ag or bg |

## Value

production as MAgPIE object (unit depends on attributes)

### Author(s)

Benjamin Leon Bodirsky

### See Also

`reportProduction`, `demand`

### Examples

```
## Not run:
x <- production(gdx)

## End(Not run)
```

---

## Residues *Residues*

### Description

reads various crop residue (carbon) outputs out of a MAgPIE gdx file

### Usage

```
Residues(
  gdx,
  level = "regglo",
  products = "kres",
  waterAggr = TRUE,
  output = "all"
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) |
| `products` | Selection of products (either "kcr" or "kres") |
| `waterAggr` | Aggregate irrigated and non-irriagted production or not (boolean). |
| `output` | Switch between different outputs: "biomass", "fieldBalance", "resDemand", all |

### Value

production as MAgPIE object (unit depends on attributes)

### Author(s)

Kristine Karstens, Michael Crawford

## See Also

ResidueBiomass

## Examples

```
  ## Not run:
    x <- Residues(gdx)

## End(Not run)
```

---

# ResidueUsage *ResidueUsage*

---

## Description

reads Crop Residue Usage out of a MAgPIE gdx file

## Usage

```
ResidueUsage(
  gdx,
  level = "reg",
  dir = ".",
  products = "kcr",
  product_aggr = FALSE,
  attributes = "dm",
  water_aggr = TRUE,
  spamfiledirectory = ""
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |
| `products` | Selection of products (either by naming products, e.g. "tece", or naming a set,e.g."kcr") |
| `product_aggr` | aggregate over products or not. Usually boolean, but here also the value "kres" is allowed, which provides kcr aggregated to kres |
| `attributes` | dry matter: Mt ("dm"), gross energy: PJ ("ge"), reactive nitrogen: Mt ("nr"), phosphor: Mt ("p"), potash: Mt ("k"), wet matter: Mt ("wm"). Can also be a vector. |
| `water_aggr` | aggregate irrigated and non-irriagted production or not (boolean). |
| `spamfiledirectory` | deprecated. please use `dir` instead |

### Value

production as MAgPIE object (unit depends on attributes)

### Author(s)

Kristine Karstens, Michael Crawford

### See Also

`ResidueBiomass`

### Examples

```
  ## Not run:
    x <- ResidueUsage(gdx)

## End(Not run)
```

---

## `RotationLength` *RotationLength*

### Description

reads rotation length out of a MAgPIE gdx file

### Usage

```
RotationLength(gdx, file = NULL, level = "regglo")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "cell", "reg" (regional), "glo" (global), "regglo" (regional and global) or any secdforest aggregation level defined in superAggregate |

### Details

Forest rotation length

### Value

Forest rotation length

## Author(s)

Abhijeet Mishra

## Examples

```
  ## Not run:
    x <- RotationLength(gdx)

## End(Not run)
```

---

`ruralDemandShares` *ruralDemandShares*

## Description

reports rural demand shares based on local consumption

## Usage

```
ruralDemandShares(
  gdx,
  type = "tradOnly",
  level = "reg",
  product_aggr = TRUE,
  file = NULL
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `type` | Type of ratio that should be calculated<br>• `all`: How much rural & trad demand as a share of all demand is satisfied locally<br>• `tradOnly`: How much rural & trad demand as a share of rural & trad demand is satisfied locally<br>• `potential`: How much total gridded demand is potentially satisfied by gridded production |
| `level` | spatial aggregation to report employment ("reg", "glo" or "regglo") |
| `product_aggr` | sum over products if TRUE |
| `file` | a file name the output should be written to using write.magpie |

## Value

share of food consumed locally

## Author(s)

David M Chen

## Examples

```
## Not run:
x <- localDemandShares(gdx)

## End(Not run)
```

---

`Seed` *Seed*

---

## Description

Calculates MAgPIE demand for Seed out of a gdx file

## Usage

```
Seed(gdx, level = "reg", attributes = "dm", dir = ".", spamfiledirectory = "")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | Level of regional aggregation ("reg", "glo", "regglo") |
| `attributes` | dry matter: Mt ("dm"), gross energy: PJ ("ge"), reactive nitrogen: Mt ("nr"), phosphor: Mt ("p"), potash: Mt ("k"), wet matter: Mt ("wm"). Can also be a vector. |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |
| `spamfiledirectory` | deprecated. please use `dir` instead |

## Details

Demand definitions are equivalent to FAO CBS categories

## Value

demand as MAgPIE object (Unit depends on attributes)

## Author(s)

Benjamin Leon Bodirsky

### Examples

```
  ## Not run:
    x <- demand(level="regglo", products="kcr")

## End(Not run)
```

---

## SOM *SOM*

### Description

Calculates soil organic carbon stock size based on a MAgPIE gdx file

### Usage

```
SOM(
  gdx,
  file = NULL,
  type = "stock",
  reference = "actual",
  level = "reg",
  noncrop_aggr = TRUE,
  dir = ".",
  spamfiledirectory = ""
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `type` | "stock" (default) for absoulte values, "density" for per hectar values |
| `reference` | default is "actual" (cshare in actual carbon stocks). Other option is "target" (cshare in target carbon stocks). |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `noncrop_aggr` | aggregate non cropland types to 'noncropland' (if FALSE all land types of pools59 will be reported) |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |
| `spamfiledirectory` | deprecated. please use `dir` instead |

### Value

A MAgPIE object containing som values

## Author(s)

Kristine Karstens

## Examples

```
## Not run:
x <- SOM(gdx)

## End(Not run)
```

---

`SOM2` *SOM2*

## Description

Calculates soil organic carbon stock size based on a MAgPIE gdx file (for threepool realization)

## Usage

```
SOM2(gdx, type = "stock", level = "regglo", noncropAggr = TRUE)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `type` | "stock" (default) for absoulte values, "density" for per hectar values |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) |
| `noncropAggr` | aggregate non cropland types to 'noncropland' (if FALSE all land types of pools59 will be reported) |

## Value

A MAgPIE object containing som values

## Author(s)

Kristine Karstens

## Examples

```
## Not run:
  x <- SOM2(gdx)

## End(Not run)
```

---

## `submitCalibration` *submitCalibration*

---

### Description

Submits Calibration Factors of current run to calibration archive. Currently covers calibration factors for yields and land conversion costs. This is useful to make runs more comparable to each other. The function can be also used as part of a script running a collection of runs.

### Usage

```
submitCalibration(
  name,
  file = c("modules/14_yields/input/f14_yld_calib.csv",
    "modules/39_landconversion/input/f39_calib.cs3"),
  archive = "/p/projects/landuse/data/input/calibration"
)
```

### Arguments

| | |
|---|---|
| `name` | name under which the calibration should be stored. Should be as self-explaining as possible. The total file name has the format calibration_<name>_<date>.tgz. |
| `file` | path to a f14_yld_calib.csv and f39_calib.cs3 (older version f39_calib.csv) file (in this order). Alternatively a fulldata.gdx file containing the calibration factors can be used. Supported file types are "csv", "cs3" and "gdx". |
| `archive` | path to the archive the calibration factors should be stored |

### Value

file name of the stored calibration factors (useful for scripts in which you might want to re-use a calibration setting at a later stage again)

### Author(s)

Jan Philipp Dietrich, Florian Humpenoeder, Patrick v. Jeetze

### Examples

```
## Not run:
fname <- submitCalibration("TestCalibration", file = "fulldata.gdx")

## End(Not run)
```

---

`superAggregateX` *superAggregateX*

---

## Description

drop-in replacement for superAggregate based on toolAggregate

## Usage

```
superAggregateX(
  data,
  aggr_type,
  level = "reg",
  weight = NULL,
  crop_aggr = FALSE
)
```

## Arguments

| | |
|---|---|
| `data` | An MAgPIE |
| `aggr_type` | Aggregation Type. Can be any function for one or two objects (data and weight) of the same size. Currently pre-supported functions: "sum","mean","weighted_mean". |
| `level` | Allowed level types are global "glo", regional "reg" and "regglo" |
| `weight` | Currently only used for weighted_mean |
| `crop_aggr` | determines whether output should be crop-specific (FALSE) or aggregated over all crops (TRUE). The method used for aggregation is set by aggr_type |

## Value

returns a MAgPIE object.

## Author(s)

Jan Philipp Dietrich

---

`surplusChange` *surplusChange*

---

## Description

calculates aggregate change in economic surplus in mio.US$ based on a MAgPIE gdx files from two different scenarios.

## Usage

```
surplusChange(gdx1, gdx2, file = NULL, level = "reg", type = "consumer")
```

## Arguments

| | |
|---|---|
| `gdx1` | GDX file from benchmark scenario |
| `gdx2` | GDX file from the analyzed scenario |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `type` | Economic surplus type: "consumer" (default), "producer" or "welfare" |

## Value

A MAgPIE object containing aggregate changes in producer surplus, consumer surplus and aggregate economic welfare between an analyzed scenario and a benchmark scenario, in million $US.

## Author(s)

Miodrag Stevanovic

## Examples

```
  ## Not run:
    x <- surplusChange(gdx1, gdx2)

## End(Not run)
```

---

# `tau` *tau*

## Description

Calculates Landuse intensity indicator tau based on a MAgPIE gdx file

## Usage

```
tau(
  gdx,
  file = NULL,
  level = "reg",
  start_value = FALSE,
  digits = 4,
  prev_year = "y1985",
  type = "crop"
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `start_value` | If TRUE, the initial values are added under the year `prev_year` |
| `digits` | The result will be rounded to this number of digits |
| `prev_year` | Year to store the initialization tau information in |
| `type` | type of tc 'pastr' or 'crop'; or "both" if both are needed |

## Value

A MAgPIE object containing tau values (index)

## Author(s)

Jan Philipp Dietrich

## Examples

```
## Not run:
x <- tau(gdx)

## End(Not run)
```

---

# `taxRevenueRotations` *taxRevenueRotations*

## Description

calculates taxes of crop rotations as difference between the selected scenario and the baseline scenario that shall capture the internalized incentives for crop rotations.

## Usage

```
taxRevenueRotations(
  gdx,
  file = NULL,
  level = "regglo",
  dir = ".",
  penalty = "onlyTaxRevenue"
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | aggregation level, reg, glo or regglo |
| `dir` | spamfiledirectory |
| `penalty` | "OnlyTaxRevenue" provides the tax Revenues from a rotation tax/subsidy. "OnlyInternalizedServices" provides the penalty by foregone Ecosystem Services, the part of the externality which is internalized by the farmer independent of the tax. "FullPenalty" provides the sum of both, which is what the model sees. |

## Author(s)

Benjamin Leon Bodirsky

## Examples

```
## Not run:
x <- wageRent(gdx)

## End(Not run)
```

---

# tc *tc*

## Description

Calculates TC rates based on a MAgPIE gdx file

## Usage

```
tc(
  gdx,
  file = NULL,
  level = "reg",
  annual = TRUE,
  avrg = FALSE,
  baseyear = 1995,
  type = "crop"
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `annual` | If TRUE, annual values are reported. If FALSE, the values for the whole timestep are reported. If FALSE, avrg has no effect |
| `avrg` | If FALSE the annual tc rates of the current period are returned, otherwise the average annual tc rate for the period tbase to tn is returned. tbase defaults to the first timestep (see baseyear) |
| `baseyear` | Determines the base year timestep for annual tc calculation. Average tc rates for later timesteps are calculated with respect to baseyear. No tc rates for timesteps before baseyear are returned) |
| `type` | type of tc 'pastr' or 'crop' |

## Value

A MAgPIE object containing tc rates. Annual ones if annual=TRUE, for the whole timestep if annual=FALSE.

## Author(s)

Jan Philipp Dietrich

## Examples

```
## Not run:
x <- tc(gdx)

## End(Not run)
```

---

# `Timber` *Timber*

## Description

reads timber demand out of a MAgPIE gdx file

## Usage

```
Timber(gdx, file = NULL, level = "regglo")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "cell", "reg" (regional), "glo" (global), "regglo" (regional and global) or any secdforest aggregation level defined in superAggregate |

### Details

Forest demandfor timber production

### Value

Forest demandfor timber production

### Author(s)

Abhijeet Mishra

### Examples

```
  ## Not run:
    x <- Timber(gdx)

## End(Not run)
```

---

## `TimberDemand` *TimberDemand*

### Description

reads timber demand out of a MAgPIE gdx file

### Usage

```
TimberDemand(gdx, file = NULL, level = "regglo")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "cell", "reg" (regional), "glo" (global), "regglo" (regional and global) or any secdforest aggregation level defined in superAggregate |

### Details

Forest demandfor timber production

## Value

Forest demandfor timber production

## Author(s)

Abhijeet Mishra

## Examples

```
  ## Not run:
    x <- TimberDemand(gdx)

## End(Not run)
```

---

`TimberDemandVolumetric`

# *TimberDemandVolumetric*

## Description

reads timber demand out of a MAgPIE gdx file

## Usage

```
TimberDemandVolumetric(gdx, file = NULL, level = "regglo")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "cell", "reg" (regional), "glo" (global), "regglo" (regional and global) or any secdforest aggregation level defined in superAggregate |

## Details

Forest demandfor timber production

## Value

Forest demandfor timber production

## Author(s)

Abhijeet Mishra

## Examples

```
  ## Not run:
    x <- TimberDemandVolumetric(gdx)

## End(Not run)
```

---

`TimberProductionVolumetric`

## *TimberProductionVolumetric*

### Description

reads timber production out of a MAgPIE gdx file

### Usage

```
TimberProductionVolumetric(
  gdx,
  file = NULL,
  level = "regglo",
  sumProduct = FALSE,
  sumSource = TRUE
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "cell", "reg" (regional), "glo" (global), "regglo" (regional and global) or any secdforest aggregation level defined in superAggregate |
| `sumProduct` | sum over wood and woodfuel (TRUE/FALSE) |
| `sumSource` | sum over timber sources: timber plantations, primary forest, secondary forest and non-forest land (woodfuel only) (TRUE/FALSE) |

### Details

Annual timber production from timber plantations, primary forest, secondary forest and non-forest land (woodfuel only). Converted from mio. ton DM per year to mio. m3 per year using volumetric conversion factors.

### Value

Timber production in mio. m3 per year

## Author(s)

Abhijeet Mishra, Florian Humpenoeder

## Examples

```
## Not run:
x <- TimberProductionVolumetric(gdx)

## End(Not run)
```

---

# `timePeriods` *timePeriods*

## Description

Calculates MAgPIE time period lengths between each two timesteps

## Usage

```
timePeriods(gdx)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |

## Value

magpie time periods as MAgPIE object as a number of years

## Author(s)

Mishko Stevanovic

## Examples

```
  ## Not run:
    x <- timePeriods(gdx=gdx)

## End(Not run)
```

---

## `totalHoursWorked` *totalHoursWorked*

### Description

returns total hours worked per year in crop+livestock production from MAgPIE results

### Usage

```
totalHoursWorked(gdx, level = "reg", file = NULL)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `level` | spatial aggregation to report employment ("reg", "glo", or "regglo") |
| `file` | a file name the output should be written to using write.magpie |

### Value

total hours worked in agriculture per year

### Author(s)

Debbora Leip

### Examples

```
## Not run:
x <- totalHoursWorked(gdx)

## End(Not run)
```

---

## `trade` *trade*

### Description

Calculates MAgPIE trade or self-sufficiencies out of a gdx file

## Usage

```
trade(
  gdx,
  file = NULL,
  level = "reg",
  products = "k_trade",
  product_aggr = FALSE,
  attributes = "dm",
  weight = FALSE,
  relative = FALSE,
  type = "net-exports"
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation ("reg", "glo", "regglo") |
| `products` | Selection of products (either by naming products, e.g. "tece", or naming a set,e.g."kcr") |
| `product_aggr` | aggregate over products or not (boolean) |
| `attributes` | dry matter: Mt ("dm"), gross energy: PJ ("ge"), reactive nitrogen: Mt ("nr"), phosphor: Mt ("p"), potash: Mt ("k"), wet matter: Mt ("wm"). Can also be a vector. |
| `weight` | in case relative=T also the weighting for the self sufficiencies is provided as it is an intensive parameter |
| `relative` | if relative=TRUE, self sufficiencies are reported, so the amount of production divided by domestic demand |
| `type` | exports-imports ("net-exports"), gross imports ("imports") or gross exports ("exports"); only valid if relative=FALSE |

## Details

Trade definitions are equivalent to FAO CBS categories

## Value

trade (production-demand) as MAgPIE object; unit depends on attributes

## Author(s)

Benjamin Leon Bodirsky, Florian Humpenoeder, Mishko Stevanovic

## Examples

```
  ## Not run:
    x <- trade(gdx="fulldata.gdx", level="regglo", products="kcr")

## End(Not run)
```

---

## tradeValue *tradeValue*

---

### Description

Calculates the value of traded goods based on a gdx file

### Usage

```
tradeValue(
  gdx,
  file = NULL,
  level = "reg",
  products = "k_trade",
  product_aggr = FALSE,
  type = "net-exports",
  glo_weight = "export",
  relative = FALSE
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation ("reg", "glo", "regglo") |
| `products` | Selection of products (either by naming products, e.g. "tece", or naming a set,e.g."kcr") |
| `product_aggr` | aggregate over products or not (boolean) |
| `type` | exports-imports ("net-exports"), gross imports ("imports") or gross exports ("exports"); only valid if relative=FALSE |
| `glo_weight` | Decides the calculation of global prices. Weighting schemes are applied for estimation of global producer price. If `"export"` prices are calculated as average of regional exporters' prices, weighted by the export volumes. If `"production"` (default), prices are calculated as average of regional prices weighted by regional production. Alternatively, if `"free_trade"`, the global prices are directly taken from the shadow prices of the global trade constraint, and no averaging is performed. Alternatively, if `"constant_prices_initial"` constant 1995 global prices for each commodity are used as weight. |
| `relative` | if relative=TRUE, self sufficiencies are reported (the amount of production divided by domestic demand) |

## Value

A MAgPIE object containing the value of trade flows in Million of US dollars

## Author(s)

Misko Stevanovic, Florian Humpenoeder, Edna J. Molina Bacca

## Examples

```
## Not run:
x <- tradeValue(gdx)

## End(Not run)
```

---

# `tryList` *tryList*

## Description

Internal support function to run a list of reportings in a `tryReport` environment.

## Usage

```
tryList(..., gdx, level = "regglo")
```

## Arguments

| | |
|---|---|
| `...` | report function to be run |
| `gdx` | gdx file to report from |
| `level` | spatial level (either "regglo" for region+global or "iso" for ISO countries) |

## Author(s)

Jan Philipp Dietrich

## See Also

`tryReport`

---

`tryReport` *tryReport*

---

## Description

Internal support function to run a reporting in a try environment and properly report problems if something goes wrong without stopping the further processing in case of an error

## Usage

```
tryReport(report, width, gdx, level = "regglo", n = 1)
```

## Arguments

| | |
|---|---|
| `report` | report function to be run |
| `width` | max number of characters per line |
| `gdx` | gdx file to report from |
| `level` | spatial level (either "regglo" for region+global or "iso" for ISO countries) |
| `n` | number of parent generations to go back when catching the environment the report should get evaluated in |

## Author(s)

Jan Philipp Dietrich

---

`validation` *validation*

---

## Description

Create Validation pdf from MAgPIE output and corresponding validation.mif

## Usage

```
validation(
  gdx,
  hist,
  file = "validation.pdf",
  runinfo = NULL,
  clusterinfo = NULL,
  debug = FALSE,
  reportfile = NULL,
  scenario = NULL,
  getReport = NULL,
  ...
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `hist` | Validation data. All formats allowed which can be converted to quitte (including characters containing the path to a mif file) |
| `file` | a file name the output pdf |
| `runinfo` | (optional) Rdata object with run information |
| `clusterinfo` | (optional) RDS file or vector containing mapping information on 0.5degree between regions and cluster |
| `debug` | Switch to activate or deactivate debug mode |
| `reportfile` | file name to which a backup of the magpie reporting should be written (file ending should be ".mif"). No report written if set to NULL or if report is already provided via getReport! |
| `scenario` | scenario name used inside reportfile. Not used if reportfile is NULL. |
| `getReport` | the return value of the `getReport` function. Can be provided if available to reduce overall runtime. |
| `...` | additional arguments supplied to the validationpdf function |

## Author(s)

Jan Philipp Dietrich

## Examples

```
  ## Not run:
    validation("fulldata.gdx","validation.mif",filter="Yield")

## End(Not run)
```

---

# `ValueMaterialDemand` *ValueMaterialDemand*

## Description

calculates agricultural costs without taxes and incentives (i.e. GHG taxes and BII incentives)

## Usage

```
ValueMaterialDemand(gdx, file = NULL, level = "regglo")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | aggregation level, reg, glo or regglo |

## Author(s)

David M Chen

## Examples

```
## Not run:
x <- ValueMaterialDemand(gdx)

## End(Not run)
```

---

## VegfruitShare *VegfruitShare*

## Description

Calculates the share of fruits, vegetables and nuts in total food supply from the food demand model

## Usage

```
VegfruitShare(
  gdx,
  file = NULL,
  level = "reg",
  after_shock = TRUE,
  calibrated = TRUE,
  attributes = "kcal"
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "iso" ISO country codes, "reg" (regional), "glo" (global) |
| `after_shock` | FALSE is using the exogenous real income and the prices before a shock, TRUE is using the endogeenous real income that takes into account food price change on real income |
| `calibrated` | if FALSE, the true regression outputs are used, if TRUE the values calibrated to the start years are used |
| `attributes` | unit: kilocalories per day ("kcal"), g protein per day ("protein"). Mt reactive nitrogen ("nr"). |

## Value

magpie object with the livestock share in a region or country. Unit is dimensionsless, but value depends on chosen attribute

## Author(s)

Benjamin Bodirsky

## Examples

```
  ## Not run:
    x <- VegfruitShare(gdx)

## End(Not run)
```

---

`wageDevelopment` *wageDevelopment*

---

## Description

calculates indicator to describe wage development based on agricultural wages in MAgPIE (hourly labor costs relative to a base year)

## Usage

```
wageDevelopment(gdx, baseYear = 2000, file = NULL, level = "regglo")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `baseYear` | year relative to which the wage development should be calculated |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation ("iso", "reg", "glo", "regglo") |

## Value

MAgPIE object containing indicator on wage development

## Author(s)

Debbora Leip

## Examples

```
## Not run:
x <- wageDevelopment(gdx)

## End(Not run)
```

---

`wageRent` *wageRent*

---

### Description

calculates wage rent for exogenous wage scenarios

### Usage

```
wageRent(gdx, file = NULL, level = "regglo")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | aggregation level, reg, glo or regglo |

### Author(s)

Debbora Leip

### Examples

```
## Not run:
x <- wageRent(gdx)

## End(Not run)
```

---

`waterEFR` *waterEFR*

---

### Description

reads environmental flow requirements (as they enter MAgPIE) from a MAgPIE gdx file

### Usage

```
waterEFR(gdx, file = NULL, level = "cell", digits = 4)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | spatial level of aggregation: "cell" (cellular), "reg" (regional), "glo" (global), "regglo" (regional and global) |
| `digits` | integer. For rounding of the return values |

## Value

A MAgPIE object containing environmental flow requirements (km^3)

## Author(s)

Felicitas Beier

## Examples

```
## Not run:
x <- waterEFR(gdx)

## End(Not run)
```

---

# `waterEFVarea` *waterEFVarea*

## Description

calculates area that falls into cluster experiencing environmental flow violations from MAgPIE outputs

## Usage

```
waterEFVarea(gdx, file = NULL, level = "reg", digits = 4, dir = ".")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | spatial level of aggregation: "cell" (cellular), "reg" (regional), "glo" (global), "regglo" (regional and global), or "grid" (for disaggregated output using cropland as weight) |
| `digits` | integer. For rounding of the return values |
| `dir` | directory for weight for disaggregation |

## Value

A MAgPIE object containing the area under environmental flow violations (Mha)

## Author(s)

Felicitas Beier

## Examples

```
## Not run:
x <- waterEFVarea(gdx)

## End(Not run)
```

---

# `waterEFViolation` *waterEFViolation*

## Description

calculates environmental flow violation volume from MAgPIE outputs

## Usage

```
waterEFViolation(gdx, file = NULL, level = "reg", digits = 4, dir = ".")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | spatial level of aggregation: "cell" (cellular), "reg" (regional), "glo" (global), "regglo" (regional and global), or "grid" (for disaggregated output using cropland as weight) |
| `digits` | integer. For rounding of the return values |
| `dir` | directory for weight for disaggregation |

## Value

A MAgPIE object containing the volume of environmental flow violations (km^3)

## Author(s)

Felicitas Beier

## Examples

```
## Not run:
x <- waterEFViolation(gdx)

## End(Not run)
```

---

# `waterEFVratio` *waterEFVratio*

---

## Description

calculates ratio of environmental flow violation volume (EFV) to water environmental flow requirements (EFR) in MAgPIE.

## Usage

```
waterEFVratio(gdx, file = NULL, level = "cell", dir = ".")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | spatial level of aggregation: "cell" (cellular), "reg" (regional), "glo" (global), "regglo" (regional and global) or "grid" (grid cell) |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |

## Value

MAgPIE object

## Author(s)

Felicitas Beier

## Examples

```
  ## Not run:
    x <- waterEFVratio(gdx)

## End(Not run)
```

---

`waterStress` *waterStress*

---

## Description

calculates which areas are affected by water stress from water availability and water demand in MAgPIE. Water stress is calculated based on the proportion of water withdrawals to water availability. Thresholds based on World Resources Institute definition (https://www.wri.org/data/water-stress-country): Low stress: <10 Low-to-medium stress: 10-20 Medium to high stress: 20-40 High stress: 40-80 Extremely high stress: >80

## Usage

```
waterStress(gdx, stressRatio = 0.4, file = NULL, level = "cell")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `stressRatio` | threshold defining level of water stress (e.g. 0.2 for medium water stress, 0.4 for high water stress) |
| `file` | a file name the output should be written to using write.magpie |
| `level` | spatial level of aggregation: "cell" (cellular), "reg" (regional), "glo" (global), "regglo" (regional and global) |

## Value

MAgPIE object indicating whether location is water stressed (1) or not (0)

## Author(s)

Felicitas Beier

## Examples

```
## Not run:
x <- waterStress(gdx)

## End(Not run)
```

---

# waterStressedPopulation

*waterStressedPopulation*

---

## Description

People living in water stressed region

## Usage

```
waterStressedPopulation(
  gdx,
  file = NULL,
  level = "cell",
  dir = ".",
  absolute = TRUE
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | spatial level of aggregation: "cell" (cellular), "reg" (regional), "glo" (global), "regglo" (regional and global) or "grid" (grid cell) |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |
| `absolute` | TRUE: reports people living in water stressed region in million, FALSE: returns share of population |

## Value

MAgPIE object

## Author(s)

Felicitas Beier

## Examples

```
  ## Not run:
    x <- waterStressRatio(gdx)

## End(Not run)
```

---

`waterStressRatio` *waterStressRatio*

---

## Description

calculates water stress ratio from water availability and water demand in MAgPIE. Water stress ratio is the ratio of water withdrawals (in the growing period) to water availability (in the growing period)

## Usage

```
waterStressRatio(gdx, file = NULL, level = "cell", dir = ".")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | spatial level of aggregation: "cell" (cellular), "reg" (regional), "glo" (global), "regglo" (regional and global) or "grid" (grid cell) |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |

## Value

MAgPIE object

## Author(s)

Felicitas Beier

## Examples

```
  ## Not run:
    x <- waterStressRatio(gdx)

## End(Not run)
```

---

## `water_AAI` *water_AAI*

---

### Description

reads area actually irrigated from a MAgPIE gdx file

### Usage

```
water_AAI(gdx, file = NULL, level = "reg", dir = ".")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using `write.magpie`. See `write.magpie` for supported file types |
| `level` | spatial level of aggregation: "cell" (cellular), "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) |

### Value

A MAgPIE object containing the area actually irrigated (Mha)

### Author(s)

Stephen Wirth, Anne Biewald, Felicitas Beier

### Examples

```
## Not run:
x <- water_AEI(gdx)

## End(Not run)
```

---

## `water_AEI` *water_AEI*

---

### Description

reads area equipped for irrigation from a MAgPIE gdx file

### Usage

```
water_AEI(gdx, file = NULL, level = "reg")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | spatial level of aggregation: "cell" (cellular), "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |

### Value

A MAgPIE object containing the area equipped for irrigation (Mha)

### Author(s)

Markus Bonsch

### Examples

```
  ## Not run:
    x <- water_AEI(gdx)

## End(Not run)
```

---

## `water_avail` *water_avail*

---

### Description

reads available water from a MAgPIE gdx file

## Usage

```
water_avail(
  gdx,
  file = NULL,
  level = "reg",
  dir = ".",
  sources = NULL,
  sum = TRUE,
  digits = 4
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | spatial level of aggregation: "cell" (cellular), "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `dir` | directory for files necessary for disaggregation |
| `sources` | Vector of water sources that shall be obtained. NULL for all sources |
| `sum` | Sum the contribution of different sources (TRUE) or display them individually (FALSE) |
| `digits` | integer. For rounding of the return values |

## Value

A MAgPIE object containing the available water (km^3)

## Author(s)

Markus Bonsch, Felicitas Beier

## Examples

```
## Not run:
x <- water_avail(gdx)

## End(Not run)
```

---

## `water_efficiency` *water_efficiency*

---

### Description

reads Irrigation efficiency from a MAgPIE gdx file

### Usage

```
water_efficiency(gdx, file = NULL, level = "reg")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | spatial level of aggregation: "cell" (cellular), "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |

### Value

A MAgPIE object containing the irrigation efficiency on the requested aggregation level

### Author(s)

Markus Bonsch

### Examples

```
  ## Not run:
    x <- water_efficiency(gdx)

## End(Not run)
```

---

## `water_price` *water_price*

---

### Description

reads water prices from a MAgPIE gdx file

## Usage

```
water_price(
  gdx,
  file = NULL,
  level = "reg",
  weight = "value",
  index = FALSE,
  index_baseyear = 2005,
  digits = 4
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | spatial level of aggregation: "cell" (cellular), "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `weight` | For determining weights to use for generating water prices at levels beyond 'cellular'. Takes "value" and "quantity". "value" sums regional weights by value of water per cluster, "quantity" sums regional weight by qty of water per cluster |
| `index` | FALSE (default) or TRUE |
| `index_baseyear` | baseyear to use for index calculation (only used if index=TRUE) |
| `digits` | integer. For rounding of the return values |

## Value

A MAgPIE object containing the water shadow prices (US Dollar/cubic metre).

## Author(s)

Markus Bonsch, Vartika Singh, Miodrag Stevanovic

## Examples

```
  ## Not run:
    x <- water_price(gdx)

## End(Not run)
```

---

## `water_usage` *water_usage*

---

### Description

reads area usage from a MAgPIE gdx file

### Usage

```
water_usage(
  gdx,
  file = NULL,
  level = "reg",
  users = NULL,
  sum = FALSE,
  seasonality = "total",
  abstractiontype = "withdrawal",
  digits = 4,
  dir = "."
)
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | spatial level of aggregation: "grid" (grid-cell) "cell" (cellular), "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in gdxAggregate |
| `users` | NULL or "sectors" or "kcr" or "kli". If NULL, all sectors including crop-wise water use and livestock will be obtained. If sectors, will only report for high-level sectors - agriculture, industry, electricity, domestic, ecosystem. Sum is applicable only in the case of sectors |
| `sum` | determines whether output should be sector specific (FALSE) or aggregated over all sectors (TRUE) |
| `seasonality` | water usage time of the year. options: "grper" (growing period) or "total" (entire year). Note: currently only implemented for non-agricultural water usage. |
| `abstractiontype` | water usage abstraction type: "withdrawal" or "consumption" |
| `digits` | integer. For rounding of the return values |
| `dir` | for gridded outputs: magpie output directory which contains a mapping file (rds) for disaggregation |

### Value

A MAgPIE object containing the water usage (km^3/yr)

### Author(s)

Markus Bonsch, Vartika Singh, Felicitas Beier

### Examples

```
## Not run:
x <- water_usage(gdx)

## End(Not run)
```

---

## `woodHarvestArea` *woodHarvestArea*

---

### Description

Reads wood harvest area separated by source (primforest, secdforest, forestry, other) and age classes from a gdx. The data is on cluster level and the unit is Mha per year.

### Usage

```
woodHarvestArea(gdx)
```

### Arguments

| | |
|---|---|
| `gdx` | A fulldata.gdx of a magpie run, usually with endogenous forestry enabled |

### Value

A magpie object with the following dimensions: region, id, year, source, ageClass

### Author(s)

Pascal Sauer

---

## `woodProduction` *woodProduction*

---

### Description

Reads roundwood and fuelwood production/harvest data separated by source (primforest, secdforest, forestry, other) from a gdx. The data is on cluster level and the unit is Petagram (= mio. t) dry matter per year (Pg DM yr-1).

### Usage

```
woodProduction(gdx)
```

## Arguments

| | |
|---|---|
| `gdx` | A fulldata.gdx of a magpie run, usually with endogenous forestry enabled |

## Value

A magpie object with the following dimensions: region, id, year, source, woodType

## Author(s)

Pascal Sauer

---

# `yields` *yields*

## Description

Calculates crop yields based on a MAgPIE gdx file

## Usage

```
yields(
  gdx,
  file = NULL,
  level = "reg",
  products = "kcr",
  product_aggr = F,
  attributes = "dm",
  water_aggr = T
)
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation; "reg" (regional), "glo" (global), "regglo" (regional and global) or any other aggregation level defined in superAggregate |
| `products` | Selection of products (either by naming products, e.g. "tece", or naming a set,e.g."kcr"), also including "pasture" |
| `product_aggr` | aggregate over products or not (boolean) |
| `attributes` | dry matter: Mt/ha ("dm"), gross energy: PJ/ha ("ge"), reactive nitrogen: Mt/ha ("nr"), phosphor: Mt/ha ("p"), potash: Mt/ha ("k"), wet matter: Mt/ha ("wm"). Can also be a vector. |
| `water_aggr` | aggregate irrigated and non-irriagted production or not (boolean). |

## Value

crop yield as MAgPIE object (unit depends on attributes)

## Author(s)

Florian Humpenoeder

## See Also

`reportYields`

## Examples

```
  ## Not run: 
    x <- yields(gdx)
  
## End(Not run)
```

---

# `YieldsCropCalib` *YieldsCropCalib*

## Description

Reads potential yields after calibration

## Usage

```
YieldsCropCalib(gdx, file = NULL, level = "cell", dir = ".")
```

## Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation |
| `dir` | directory for disaggregation to grid level |

## Value

A MAgPIE object containing values of potential yields after the calibration routines

## Author(s)

Edna Molina Bacca

## Examples

```
## Not run:
x <- YieldsCropCalib(gdx)

## End(Not run)
```

---

## `YieldsCropRaw` *YieldsCropRaw*

### Description

Reads potential yields after calibration

### Usage

```
YieldsCropRaw(gdx, file = NULL, level = "cell")
```

### Arguments

| | |
|---|---|
| `gdx` | GDX file |
| `file` | a file name the output should be written to using write.magpie |
| `level` | Level of regional aggregation |

### Value

A MAgPIE object containing values of potential yields after the calibration routines

### Author(s)

Edna Molina Bacca

### Examples

```
## Not run:
x <- YieldsCropRaw(gdx)

## End(Not run)
```

# Index