

# Package ‘decctools’

August 4, 2014

**Type** Package

**Author** James Keirstead <j.keirstead@imperial.ac.uk>

**Maintainer** James Keirstead <j.keirstead@imperial.ac.uk>

**Version** 0.2.0

**Date** 2014-08-04

**License** GPL-2

**Title** Get energy data from the UK Dept of Energy and Climate Change

**Description** decctools provides functions for retrieving energy statistics from the United Kingdom Department of Energy and Climate Change and related data sources. The current version focuses on total final energy consumption statistics at the local authority, MSOA, and LSOA geographies. Methods for calculating the generation mix of grid electricity and its associated carbon intensity are also provided.

**Depends** XLConnect

**Imports** stringr, XML, reshape2, RCurl, lubridate, plyr

**LazyData** false

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2014-08-04 16:43:28

## R topics documented:

carbon_intensities . . . . .	2
clean_decc_data . . . . .	3
decctools . . . . .	3
get_geo_lookup . . . . .	3
get_grid_carbon . . . . .	4

get_grid_mix . . . . .	5
get_LAD_data . . . . .	6
get_LAD_metadata . . . . .	7
get_LAD_years . . . . .	7
get_last_date . . . . .	8
get_LSOA_data . . . . .	8
get_LSOA_metadata . . . . .	9
get_LSOA_years . . . . .	9
get_MSOA_data . . . . .	10
get_MSOA_metadata . . . . .	10
get_MSOA_years . . . . .	11
get_package_name . . . . .	11
get_params_list . . . . .	12
get_remote_file . . . . .	12
get_SOA_data . . . . .	13
get_SOA_metadata . . . . .	13
get_SOA_years . . . . .	14
is_urban . . . . .	14
LAD_metadata . . . . .	15
params . . . . .	17
parse_raw_SOA_data . . . . .	18
process_tab . . . . .	18
scotland_igz . . . . .	19
validate_directory . . . . .	19
validate_SOA_level . . . . .	20

<b>Index</b>	<b>21</b>
--------------	-----------

---

carbon_intensities	<i>Carbon intensity data</i>
--------------------	------------------------------

---

### Description

A data frame giving the carbon intensity of different electricity generating sources including electricity flows from interconnections with other European countries. The values are measured in g CO<sub>2</sub>/kWh.

### Format

A data frame with 13 rows and 2 columns

### Source

Data come from <https://www.gov.uk/government/organisations/department-of-energy-climate-change/series/fuel-mix-disclosure-data-tables>, <http://www.earth.org.uk/note-on-UK-grid-CO2-intensity-variation> html, and <https://www.gov.uk/government/publications/2012-greenhouse-gas-conversion-factors-for-comparison>

---

clean_decc_data	<i>Cleans the raw DECC energy consumption data for easier use</i>
-----------------	---

---

**Description**

Cleans the raw DECC energy consumption data for easier use

**Usage**

```
clean_decc_data(df)
```

**Arguments**

df                    the raw DECC data frame from the spreadsheet

**Value**

a long database with headers by sector and fuel

---

decctools	<i>decctools</i>
-----------	------------------

---

**Description**

Provides easy access to United Kingdom energy statistics for R. Most of the underlying data comes from the Department of Energy and Climate Change website, <http://www.decc.gov.uk>. A working internet connection is needed for almost all of the functions.

**See Also**

[get\\_LAD\\_data](#), [get\\_MSOA\\_data](#), [get\\_LSOA\\_data](#), [get\\_geo\\_lookup](#), [get\\_grid\\_mix](#), [get\\_grid\\_carbon](#)

---

get_geo_lookup	<i>Gets a lookup table to match IDs across geographies</i>
----------------	--

---

**Description**

Provides a lookup table giving the local authority name and corresponding IDs for LAD, MSOA, and LSOA geographies.

**Usage**

```
get_geo_lookup()
```

**Value**

a data frame

**Note**

As SOA energy data are not available for Northern Ireland, this method only returns a lookup for England, Wales, and Scotland. Also these lookup tables are based on the 2011 geographies. When fetching data for other years, users may find that codes don't align. The recommended strategy is to merge on the name where possible.

---

get_grid_carbon	<i>Calculates the carbon intensity of electricity generation</i>
-----------------	--

---

**Description**

Calculates the carbon intensity of electricity generation in the UK between specified dates.

**Usage**

```
get_grid_carbon(start, end)
```

**Arguments**

start	the start date for retrieving data.
end	the end date for retrieving data

**Value**

a data frame with a datetime stamp and average carbon intensity of electricity measured in kg CO2/kWh

**See Also**

[get\\_grid\\_mix](#) and [carbon\\_intensities](#) for data sources

**Examples**

```
# These require a working internet connection
start <- "2010-01-01"
end <- "2010-01-03"
# Gets grid carbon for 1--3 January 2010
carbon <- get_grid_carbon(start, end)
```

---

get_grid_mix	<i>Gets mix of fuel sources in UK electricity</i>
--------------	---

---

**Description**

Gets the mix of fuel sources used to generate UK grid electricity between specified dates. Valid dates are yyyy-mm-dd and must be between 2009-01-01 and the current day; the start must come before the end date.

**Usage**

```
get_grid_mix(start, end)
```

**Arguments**

start	the start date for retrieving data.
end	the end date for retrieving data

**Value**

a data frame with a datetime stamp and average power generated by fuel type in MW

**Source**

The underlying data come from BM Reports, [http://www.bmreports.com/bsp/bsp\\_home.htm](http://www.bmreports.com/bsp/bsp_home.htm), and are used under license from Elexon (see disclaimer below). However the BM Reports website is hard to use and so this function uses archived Elexon data compiled by the charity Renewable Energy Foundation and available at no cost at <http://www.ref.org.uk/fuel/>.

ELEXON does not and shall not be deemed to warrant or guarantee or make any representation (whether expressly or impliedly) as to the title, completeness, accuracy, reliability, freeness from error, method of preparation, merchantability or fitness for the purpose or for any particular purpose (whether or not any such purpose has been notified to ELEXON) of the Licensed Data and/or any use of or dealing with it. ELEXON shall have no liability or responsibility whatsoever or howsoever arising (whether directly or indirectly) as a result of or in connection therewith, including in respect of each and all loss of profits, loss of revenue, loss of goodwill, loss of contracts, loss of business or production, indirect loss, economic or consequential loss, claims, demands, proceedings, actions, losses, liabilities, damages, costs, charges and expenses (whether or not any of the foregoing are foreseeable)

**Examples**

```
# These require a working internet connection
start <- "2010-01-01"
end <- "2010-01-03"

# Gets data for 1--3 January 2010
data <- get_grid_mix(start, end)
```

---

get\_LAD\_data

*Get LAD energy consumption data*


---

### Description

Gets LAD (Local Administrative District) energy data from the DECC website, <http://www.decc.gov.uk>. Since the format of these files tends to change just enough to be irritating for dependent code, there is also an option that allows you to cache a local copy. Note that the total values do not contain the unallocated fuel values, which represent approximately 0.3 demand.

### Usage

```
get_LAD_data(year = max(get_LAD_years()), sector = "total",
             fuel = "total", id, dir)
```

### Arguments

year	the year to fetch. If not specified, then the most recent year is retrieved. Only single years currently supported
sector	a vector of economic sectors to fetch. Valid values are 'domestic', 'industrial', 'transport', 'total', 'all'. Default = 'total'
fuel	a vector of fuel types to fetch. Valid values are 'coal', 'manufactured', 'petrol', 'gas', 'electricity', 'bioenergy', 'total', 'all'. Default = 'total'. 'All' includes the total as well.
id	the unique id of the LAD to fetch. If not specified, then all LADs are retrieved. ##'
dir	an optional directory in which to cache the data

### Value

a long data frame with the requested data. The 'energy' column is measured in GWh.

### Examples

```
## Not run:
# Gets energy data for total fuels and sectors for most recent year
lad_data <- get_LAD_data()

## End(Not run)

## Not run:

# Gets energy data for electricity and gas use in the domestic
sector in the most recent year # Depending on the status of DECC
servers, this can sometimes fail. In which case, an empty # data
frame is returned.
```

```
df <- get_LAD_data(sector="domestic", fuel=c("electricity",  
"gas"))  
  
## End(Not run)
```

---

get\_LAD\_metadata      *Gets metadata to describe LADs*

---

### **Description**

Gets metadata to describe the local authority districts

### **Usage**

```
get_LAD_metadata(dir)
```

### **Arguments**

dir                    an optional directory in which to save a copy of the metadata

### **Value**

a data frame giving metadata for each LAD. See [LAD\\_metadata](#)

---

get\_LAD\_years      *Get the years for which LAD data is available*

---

### **Description**

Get the years for which LAD data is available

### **Usage**

```
get_LAD_years()
```

### **Value**

a vector of years

---

get_last_date	<i>Gets the most recent date for available data</i>
---------------	---

---

**Description**

Gets the date of the last update to the REF fuels mix database.

**Usage**

```
get_last_date()
```

**Value**

a Date object

---

get_LSOA_data	<i>Get LSOA energy consumption data</i>
---------------	---

---

**Description**

This function fetches LSOA (Lower Super Output Area) data from the DECC website.

**Usage**

```
get_LSOA_data(year = max(get_LSOA_years()), fuel = c("electricity", "gas"),
  sector = "domestic", id, dir)
```

**Arguments**

year	the years of data to fetch. The default is the most recent year.
sector	a vector of economic sectors to fetch. For LSOA data, only 'domestic' is allowed.
fuel	the fuel type to fetch. Valid values are 'electricity', 'gas'
id	a vector of LSOA ids to fetch. If not specified, then all LSOAs are retrieved.
dir	an optional directory in which to store a copy of the data

**Value**

a long data frame with the requested data. The 'energy' column is measured in GWh.

**Examples**

```
## Not run:
lsoa_data <- get_LSOA_data() # Gets all data

## End(Not run)
```



---

get_LSOA_metadata	<i>Gets metadata for all LSOAs</i>
-------------------	------------------------------------

---

**Description**

Gets the socio-demographic data associated with each Lower Super Output Area (LSOA). These data only cover England and Wales.

**Usage**

```
get_LSOA_metadata(dir)
```

**Arguments**

dir                    an (optional) directory in which to save the downloaded data

**Value**

a data frame with the LSOA id code, population, area (in hectares), and number of households

**Source**

<https://www.gov.uk/government/statistical-data-sets/socio-economic-data-for-mlsoa-igz-and-llsoa-el>

---

get_LSOA_years	<i>Gets the years for which LSOA data are available</i>
----------------	---

---

**Description**

Gets the years for which LSOA data are available

**Usage**

```
get_LSOA_years()
```

**Value**

a numeric vector of valid years

---

get_MSOA_data	<i>Get MSOA energy consumption data</i>
---------------	---

---

### Description

This function fetches MSOA (Middle Super Output Area) data from the DECC website. Since the format of these files tends to change just enough to be irritating for dependent code, there is also an option that allows you to cache a local copy.

### Usage

```
get_MSOA_data(year = max(get_MSOA_years()), sector = c("domestic",
  "nondomestic"), fuel = c("electricity", "gas"), id, dir)
```

### Arguments

year	the year for which you want data. Defaults to the most recent year available.
sector	the economic sector to fetch. Valid values are 'domestic', 'nondomestic'
fuel	the fuel type to fetch. Valid values are 'electricity', 'gas'
id	the unique id of the MSOA to fetch. If not specified, then all MSOAs are retrieved.
dir	an optional directory in which to save a copy of the data

### Value

a long data frame with the requested data. The 'energy' column is measured in GWh.

### Examples

```
## Not run:
msoa_data <- get_MSOA_data() # Gets all data

## End(Not run)
```

---

get_MSOA_metadata	<i>Gets metadata for all MSOAs (including Scottish IGZs)</i>
-------------------	--

---

### Description

Gets the socio-demographic data associated with each Middle Super Output Area (MSOA).

### Usage

```
get_MSOA_metadata(dir)
```

**Arguments**

`dir` an (optional) directory in which to save the downloaded data

**Value**

a data frame with the MSOA id code, population, area (in hectares), and number of households

**Source**

<https://www.gov.uk/government/statistical-data-sets/socio-economic-data-for-mlsoa-igz-and-llsoa-el>

---

<code>get_MSOA_years</code>	<i>Gets the years for which MSOA data are available</i>
-----------------------------	---

---

**Description**

Gets the years for which MSOA data are available

**Usage**

```
get_MSOA_years()
```

**Value**

a numeric vector of valid years

---

<code>get_package_name</code>	<i>Gets the name of this package</i>
-------------------------------	--------------------------------------

---

**Description**

Gets the name of this package

**Usage**

```
get_package_name()
```

**Value**

a character string

---

get_params_list	<i>Builds a master set of parameters for SOA data</i>
-----------------	---

---

**Description**

Creates a list of various parameters needed to download and extract SOA data from the DECC website.

**Usage**

```
get_params_list(level)
```

**Arguments**

level	one of "LSOA" or "MSOA" specifying the output area level
-------	--

**Value**

a list containing the parameters necessary to read each LSOA data file

---

get_remote_file	<i>Gets a file from a remote URL</i>
-----------------	--------------------------------------

---

**Description**

Since the URLs of UK government data appear to change quite frequently, the method will automatically cache a local copy.

**Usage**

```
get_remote_file(url, dir, update_cache = FALSE)
```

**Arguments**

url	the address of the file to download
dir	the directory to store the cached file in. If undefined, it defaults to \$TEMP/decctools.
update_cache	boolean. Should the cached file be overwritten? Default = FALSE

**Value**

the file name (local path)

---

get_SOA_data	<i>Get SOA energy consumption data</i>
--------------	--

---

**Description**

Gets SOA (Super Output Area) energy consumption data from the DECC website. You can select which SOA level, sector, year, and fuel type to fetch, as well as specifying a directory for local caching.

**Usage**

```
get_SOA_data(level, year = max(get_SOA_years()), sector = c("domestic",
  "nondomestic"), fuel = c("electricity", "gas"), id, dir)
```

**Arguments**

level	either "LSOA" or "MSOA"
year	the year for which you want data. Defaults to the most recent year available.
sector	the economic sector to fetch. Valid values are 'domestic', 'nondomestic'
fuel	the fuel type to fetch. Valid values are 'electricity', 'gas'
id	the unique id of the SOA to fetch. If not specified, then all SOAs are retrieved.
dir	an optional directory in which to save a copy of the data

**Value**

a long data frame with the requested data. The 'energy' column is measured in GWh.

---

get_SOA_metadata	<i>Gets metadata for Super Output Areas</i>
------------------	---

---

**Description**

Gets the socio-demographic data associated with each Super Output Area.

**Usage**

```
get_SOA_metadata(level, dir)
```

**Arguments**

level	one of "LSOA" (default) or "MSOA" specifying the output area level
dir	an (optional) directory in which to save the downloaded data

**Value**

a data frame with the SOA id code, population, area (in hectares), and number of households

**Source**

<https://www.gov.uk/government/statistical-data-sets/socio-economic-data-for-mlsoa-igz-and-llsoa-el>

---

get_SOA_years	<i>Gets the years for which SOA data are available</i>
---------------	--

---

**Description**

Gets the years for which SOA data are available

**Usage**

```
get_SOA_years(level = "LSOA")
```

**Arguments**

level                    one of "LSOA" (default) or "MSOA" specifying the output area level

**Value**

a numeric vector of valid years for level

---

is_urban	<i>Checks if a specified local authority is urban</i>
----------	---

---

**Description**

Checks if a local authority is urban according to a user specified classification. See the details for more, well, detail.

**Usage**

```
is_urban(lad, urban = c("MU", "LU", "OU"))
```

**Arguments**

lad                    a character vector a LAD ids (post-2011 format)

urban                  a character vector specifying which of the above classifications to consider as a urban

## Details

The urban classifications are described at <http://www.ons.gov.uk/ons/guide-method/geography/products/area-classifications/rural-urban-definition-and-la/rural-urban-local-authority--la--classindex.html> and are summarized as follows:

- MU = Major Urban: districts with either 100,000 people or 50 per cent of their population in urban areas with a population of more than 750,000
- LU = Large Urban: districts with either 50,000 people or 50 per cent of their population in one of 17 urban areas with a population between 250,000 and 750,000
- OU = Other Urban: districts with fewer than 37,000 people or less than 26 per cent of their population in rural settlements and larger market towns
- SR = Significant Rural: districts with more than 37,000 people or more than 26 per cent of their population in rural settlements and larger market towns
- R50 = Rural-50: districts with at least 50 per cent but less than 80 per cent of their population in rural settlements and larger market towns
- Rural-80: districts with at least 80 per cent of their population in rural settlements and larger market towns

Unfortunately these classifications are only defined for England. For all other countries, assumptions have been made about which LAUs are urban (LU) or rural (SR) using the following data sources:

- Wales, <http://www.ons.gov.uk/ons/guide-method/geography/beginner-s-guide/administrative/wales/unitary-authorities/index.html>
- Scotland, <http://www.scotland.gov.uk/Publications/2004/06/19498/38788>
- Northern Ireland, <http://www.nisra.gov.uk/geography/default.asp10.htm>

## Value

a boolean vector of length(`lad`) indicating whether the corresponding local authority is urban

---

LAD\_metadata

*Descriptive data on local administrative districts*

---

## Description

A data frame providing extra information about local administrative districts (LADs) in the United Kingdom. This includes the following fields:

## Format

A data frame with 406 rows and 11 columns

## Details

- name: the name of the LAD
- country: the country in which the LAD is located
- new: the UK geography local administrative district long code (LAD11CD) ##'
- old: the UK geography local administrative district short code (LAD11CDO)
- urban\_class: the urban/rural classification
- group\_name: a descriptive label describing the character of each LAD (the Supergroup of ONS's area classifications)
- population: mid-year 2011 population estimates for each LAD
- area: the area of the LAD in hectares
- hdd: the heating degree days of the local climate
- cdd: the cooling degree days of the local climate

There are multiple subnational geographies available for the UK; please see <http://www.ons.gov.uk/ons/guide-method/geography/beginner-s-guide/index.html> for an overview. This data set tries to make it easier to handle the links between two of these geographies: local administrative districts (LADs, the UK administrative geography for local authorities) and local administrative units (LAUs, the pan-European equivalent for EuroStat).

The first five data fields, and the area field, should be self-explanatory and can be used to link other data sets, most notably the energy statistics provided by [get\\_LAD\\_data](#). However the other items require a bit of explanation.

urban\_class: These classifications are described at <http://www.ons.gov.uk/ons/guide-method/geography/products/area-classifications/rural-urban-definition-and-la/rural-urban-local-authority-index.html> and are summarized as follows:

- MU = Major Urban: districts with either 100,000 people or 50 per cent of their population in urban areas with a population of more than 750,000
- LU = Large Urban: districts with either 50,000 people or 50 per cent of their population in one of 17 urban areas with a population between 250,000 and 750,000
- OU = Other Urban: districts with fewer than 37,000 people or less than 26 per cent of their population in rural settlements and larger market towns
- SR = Significant Rural: districts with more than 37,000 people or more than 26 per cent of their population in rural settlements and larger market towns
- R50 = Rural-50: districts with at least 50 per cent but less than 80 per cent of their population in rural settlements and larger market towns
- Rural-80: districts with at least 80 per cent of their population in rural settlements and larger market towns

Unfortunately these classifications are only defined for England. For all other countries, assumptions have been made about which LAUs are urban (LU) or rural (SR) using the following data sources:

- Wales, <http://www.ons.gov.uk/ons/guide-method/geography/beginner-s-guide/administrative/wales/unitary-authorities/index.html>



- Scotland, <http://www.scotland.gov.uk/Publications/2004/06/19498/38788>
- Northern Ireland, <http://www.nisra.gov.uk/geography/default.asp10.htm>

group\_name The Office for National Statistics calculated an area classification for each local authority; based on 2001 census data; see <http://www.ons.gov.uk/ons/guide-method/geography/products/area-classifications/ns-area-classifications/index/index.html>. This field presents the group name, which is simply a descriptive label of the character of each LAD.

hdd, cdd: These fields represent annual heating and cooling degree days for each LAD; the raw data come from <http://www.eci.ox.ac.uk/research/energy/degreedays.php>. Heating degree days are assumed to be measured against a 15.5 degrees C base temperature and cooling degree days against 13 degrees C. The Oxford data set provides point estimates for 77 locations around the UK; an inverse distance weighting interpolation was performed (with  $p = 0.2$ ) to give a smooth surface of degree days across the UK and this was sampled using the LAD polygons to estimate the heating and cooling degree days in each location.

### Note

Please be aware that the LAD codes change from time to time and so if you join this data with new data sets, you should manually check that the merge has been successful. Subtle variations in name also exist so beware if merging by name.

### Source

Various; see URLs in detail above. Lookup codes from <http://www.ons.gov.uk/ons/guide-method/geography/products/names--codes-and-look-ups/names-and-codes-listings/names-and-codes-for-eurostat/index.html> (Eurostat LAU) and <http://www.ons.gov.uk/ons/guide-method/geography/products/names--codes-and-look-ups/names-and-codes-listings/names-and-codes-for-administrative-geography/index.html> for UK LADs. Population data from <http://www.ons.gov.uk/ons/rel/census/2011-census/population-and-household-estimates-for-the-united-kingdom/>.

---

params

*SOA parameters*

---

### Description

Provides a table of parameters for downloading SOA energy consumption data. Used by [get\\_SOA\\_data](#).

### Format

a data frame with 8 columns giving the parameters needed to read the Excel spreadsheets containing the energy data

---

parse\_raw\_SOA\_data      *Parses raw SOA data*

---

### Description

Parses the raw SOA data for a set of given parameters. This function deals with the fact that every DECC spreadsheet has a slightly different layout, but the extraction is the same. Namely, open up an Excel spreadsheet, get the right tab, extract the data from the right columns, tidy, and return.

### Usage

```
parse_raw_SOA_data(level, params)
```

### Arguments

level	one of "LSOA" or "MSOA" specifying the output area level
params	a list giving the function parameters including

### Value

a data frame with the energy data. NULL if not all of the required parameters are specified

---

process\_tab      *Loads data from a specified tab*

---

### Description

Loads energy data from a specified tab in the LAD workbook.

### Usage

```
process_tab(wb, y, params)
```

### Arguments

wb	the open workbook containing all of the tabs
y	the year of data to fetch
params	a list of parameters giving the target id, sector, and fuel

### Value

a data frame with the data

---

scotland_igz	<i>Scotland geography data</i>
--------------	--------------------------------

---

**Description**

Provides a lookup between Scottish districts and their Scottish Executive codes. Used internally to build the [get\\_geo\\_lookup](#) table.

**Format**

a data frame with 3 columns

---

validate_directory	<i>Validates a user-specified directory</i>
--------------------	---

---

**Description**

Ensures that a user-specified directory exists. If the argument is missing then a placeholder directory is created in \$TEMP.

**Usage**

```
validate_directory(dir)
```

**Arguments**

dir	the directory name
-----	--------------------

**Value**

the validated directory name

---

validate\_SOA\_level      *Validates a SOA level*

---

**Description**

Only one entry is allowed, either LSOA or MSOA

**Usage**

```
validate_SOA_level(level)
```

**Arguments**

level                  a character vector of unprocessed SOA level values

**Value**

a single valid level value

# Index

## \*Topic **datasets**

- carbon\_intensities, 2
- LAD\_metadata, 15
- params, 17
- scotland\_igz, 19

## \*Topic **data**

- get\_LAD\_data, 6
- get\_LSOA\_data, 8
- get\_MSOA\_data, 10
- get\_SOA\_data, 13

## \*Topic **energy**

- get\_LAD\_data, 6
- get\_LSOA\_data, 8
- get\_MSOA\_data, 10
- get\_SOA\_data, 13

- carbon\_intensities, 2, 4
- clean\_decc\_data, 3

- decctools, 3

- decctools-package (decctools), 3

- get\_geo\_lookup, 3, 3, 19
- get\_grid\_carbon, 3, 4
- get\_grid\_mix, 3, 4, 5
- get\_LAD\_data, 3, 6, 16
- get\_LAD\_metadata, 7
- get\_LAD\_years, 7
- get\_last\_date, 8
- get\_LSOA\_data, 3, 8
- get\_LSOA\_metadata, 9
- get\_LSOA\_years, 9
- get\_MSOA\_data, 3, 10
- get\_MSOA\_metadata, 10
- get\_MSOA\_years, 11
- get\_package\_name, 11
- get\_params\_list, 12
- get\_remote\_file, 12
- get\_SOA\_data, 13, 17
- get\_SOA\_metadata, 13

- get\_SOA\_years, 14

- is\_urban, 14

- LAD\_metadata, 7, 15

- params, 17
- parse\_raw\_SOA\_data, 18
- process\_tab, 18

- scotland\_igz, 19

- validate\_directory, 19

- validate\_SOA\_level, 20