

# Package ‘fcall’

November 19, 2025

**Title** Parse Farm Credit Administration Call Report Data into Tidy Data Frames

**Version** 0.1.5

**Description** Parses financial condition and performance data (Call Reports) for institutions in the United States Farm Credit System. Contains functions for downloading files from the Farm Credit Administration (FCA) Call Report archive website and reading the files into tidy data frame format. The archive website can be found at <https://www.fca.gov/bank-oversight/call-report-data-for-download>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.3

**Depends** R (>= 4.1)

**Imports** cli, dplyr, glue, purrr, rlang, stringr, tibble, tidyr, utils, waldo

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0), withr

**URL** <https://ketchbrookanalytics.github.io/fcall/>,  
<https://github.com/ketchbrookanalytics/fcall>

**BugReports** <https://github.com/ketchbrookanalytics/fcall/issues>

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Michael Thomas [aut, cre],  
Ivan Millanes [aut],  
Ketchbrook Analytics [cph, fnd]

**Maintainer** Michael Thomas <mthomas@ketchbrookanalytics.com>

**Repository** CRAN

**Date/Publication** 2025-11-19 19:00:07 UTC

## Contents

compare_files_content . . . . .	2
compare_metadata . . . . .	3
download_data . . . . .	4
file_metadata . . . . .	5
get_codes_dict . . . . .	6
process_data . . . . .	7
process_data_file . . . . .	8
process_metadata_file . . . . .	9
RCB2__AssetCodeRCB2 . . . . .	10
RCB3__DebtMaturityCode . . . . .	10
RCB__INV_CODE . . . . .	11
RCF1__LOANSTATUS . . . . .	11
RCF__LOANSTATUS . . . . .	12
RCI2B__DerivCode . . . . .	12
RCI2C__ExposureCode . . . . .	13
RCI2D__DerivRMCode . . . . .	13
RCO__ASSET_CODE . . . . .	14
RCR3__RegCapCode . . . . .	14
RCR7__RegCapCode . . . . .	15
read_data_file . . . . .	15
RID__CAP_CODE . . . . .	16
RIE1__ACLCode . . . . .	16
<b>Index</b>	<b>18</b>

---

compare\_files\_content *Compare content of a specific file between two folders*

---

### Description

compare\_files\_content() reads the content of a specified file from two folders and compares the content using the waldo::compare function. It identifies any differences in the content and returns the comparison results.

### Usage

```
compare_files_content(filename, dir1, dir2)
```

### Arguments

filename	A character string specifying the name of the file to compare.
dir1	A character string specifying the path to the first folder.
dir2	A character string specifying the path to the second folder.

**Details**

compare\_files\_content() reads the content of the specified file from both folders using readLines(), and compares the content using the waldo::compare() function.

**Value**

A list containing information about differences in the content of the specified file.

---

compare_metadata	<i>Compare FCA Call Report metadata files between two folders</i>
------------------	-------------------------------------------------------------------

---

**Description**

compare\_metadata() compares the content of the metadata files (files that start with "D\_") between two specified folders containing FCA Call Report data (from two different quarters).

**Usage**

```
compare_metadata(dir1, dir2)
```

**Arguments**

dir1	(String) The path to a folder containing FCA Call Report .TXT files for a single quarter
dir2	(String) The path to a folder containing FCA Call Report .TXT files for a (different) single quarter

**Details**

compare\_metadata() lists metadata files in each folder, identifies shared metadata files, and then compares (a) the number of files, (b) file names, (c) file order, and (d) file content (using the waldo::compare() function).

**Value**

A list containing information about differences in file names, file order, and content differences between the metadata files in dir1 and dir2

**Examples**

```
# Download data from September 2025
path_1 <- tempfile("fcadata1")
dir.create(path_1)

download_data(
  year = 2025,
  month = 9,
```

```

    dest = path_1
  )

  # Download data from September 2011
  path_2 <- tempfile("fcadata2")
  dir.create(path_2)

  download_data(
    year = 2011,
    month = 9,
    dest = path_2
  )

  compare_metadata(path_1, path_2)

```

---

download\_data

*Download data from FCA website*


---

## Description

Download data from FCA website

## Usage

```
download_data(year, month, dest, files = NULL, quiet = FALSE)
```

## Arguments

year	(Integer) The year of the Call Report (e.g., 2022)
month	(String) The month of the Call Report (e.g., "March"); you may also supply an integer (e.g., 3) representing the month numerically
dest	(String) The path to the directory where the data will be downloaded (and unzipped) into
files	(Optional) Character vector, representing the names of the files in the .zip archive to be downloaded; default is NULL, meaning all files will be downloaded
quiet	(Optional) Logical. Controls whether download progress messages are displayed in the console. Defaults to TRUE.

## Details

FCA publishes Call Report data quarterly. These .zip files are typically named "YYYYMarch.zip", "YYYYJune.zip", "YYYYSeptember.zip" and "YYYYDecember.zip" (where YYYY represents the 4-digit year). Therefore, valid values to the month argument should be limited to c(3, 6, 9, 12), unless there is an anomaly in FCA's reporting/publishing. Check <https://www.fca.gov/bank-oversight/call-report-data-for-download> to ensure the data is available for the quarter you are interested in.

**Value**

Console message informing the user where the data was successfully downloaded (and unzipped) into

**Examples**

```
path_1 <- tempfile("fcadata1")
dir.create(path_1)

download_data(
  year = 2025,
  month = "September", # using the name of the month
  dest = path_1
)

list.files(path_1)

path_2 <- tempfile("fcadata2")
dir.create(path_2)

download_data(
  year = 2025,
  month = 9, # using the month number (to refer to September)
  dest = path_2,
  # only download the following files
  files = c(
    "D_INST.TXT",
    "INST_Q202509_G20251112.TXT"
  )
)

list.files(path_2)
```

---

file\_metadata

*Descriptions for data files*

---

**Description**

Descriptions for data files

**Usage**

file\_metadata

**Format**

file\_metadata:

A data frame with 36 rows and 2 columns:

**file\_prefix** Data file root name

**description** Short description of data file

**Source**

Metadata files headers

---

get_codes_dict	<i>Retrieve dictionary of lookup codes for a specified dataset name</i>
----------------	-------------------------------------------------------------------------

---

**Description**

get\_codes\_dict() searches for an internal .rda file in the specified package and retrieves the codes dictionary based on the provided data name and naming convention. The naming convention is assumed to include the data name followed by a double underscore "\_\_".

**Usage**

```
get_codes_dict(data_name)
```

**Arguments**

data\_name      A character string specifying the data name to retrieve the codes dictionary for.

**Details**

get\_codes\_dict() uses the provided data name to construct the expected naming convention and searches for an internal .rda file in the specified package. If found, it attempts to retrieve the codes dictionary using get and returns it; otherwise, it returns NULL.

**Value**

A list with the codes dictionary (codes\_dict) and the associated variable name (codes\_varname) if found, otherwise each element will be NULL.

**Examples**

```
rcb_dict <- get_codes_dict("RCB")

# Access codes dictionary
rcb_dict$codes_dict

# Access the name of the variable that stores the codes
rcb_dict$codes_varname
```

---

`process_data`*Process FCA Call Report data in a specified folder*

---

### Description

`process_data()` reads the downloaded (and unzipped) .TXT files into tidy data frames, applying the schema from the "D\_" files to the corresponding raw comma-separated data files, as well as storing the metadata from the "D\_" files

### Usage

```
process_data(dir)
```

### Arguments

`dir` (String) The path to a folder containing FCA Call Report .TXT files for a single quarter

### Details

`process_data()` assumes that metadata and data files share a common root name (characters until the first underscore occurrence).

### Value

A list containing processed data and metadata.

### Examples

```
path <- tempfile("fcadata")
dir.create(path)

download_data(
  year = 2025,
  month = "September",
  dest = path
)

processed_data <- process_data(path)

# Access "RCB" data
processed_data$data$RCB

# Access "RCB" metadata
processed_data$metadata$RCB
```

---

process\_data\_file      *Process a data file using metadata and codes dictionary*

---

### Description

process\_data\_file() reads a data file, applies the provided metadata and codes dictionary, and organizes the data into a tidy format. The column names are determined based on the metadata scenario (e.g., "single", "single\_multiple", "single\_multiple\_single").

### Usage

```
process_data_file(file, metadata, dict = NULL)
```

### Arguments

file	(String) The path to the data file
metadata	A list containing the scenario and variable information obtained from the metadata file using <a href="#">process_metadata_file</a> .
dict	(Optional) A data frame containing codes dictionary information

### Details

process\_data\_file() processes the data file according to the metadata scenario. It handles cases where variables have multiple occurrences and organizes the data into a tidy format with appropriate column names. The function relies on the [read\\_data\\_file](#) function for the actual data reading.

### Value

A tibble containing the processed data in a tidy format

### Examples

```
path <- tempfile("fcadata")
dir.create(path)

download_data(
  year = 2025,
  month = "September",
  dest = path
)

process_data_file(
  file = file.path(path, "RCB_Q202509_G20251112.TXT"),
  metadata = process_metadata_file(file.path(path, "D_RCB.TXT")),
  dict = RCB__INV_CODE
)
```



---

process\_metadata\_file *Process metadata file to extract variable information*

---

## Description

process\_metadata\_file() reads a metadata file and extracts information about the column names, column types, decimal positions, and variable definitions.

## Usage

```
process_metadata_file(file)
```

## Arguments

file (String) The path to the metadata file.

## Details

process\_metadata\_file() processes metadata files following specific rules to handle encoding, remove unnecessary information, and extract variable details. It detects the scenario based on the occurrence of double asterisks in variable names.

## Value

A list containing the scenario (e.g., "single", "single\_multiple", "single\_multiple\_single") and a tibble with variable information.

## Examples

```
path <- tempfile("fcadata")
dir.create(path)

download_data(
  year = 2025,
  month = "September",
  dest = path
)

process_metadata_file(file.path(path, "D_RC1.TXT"))
```

---

RCB2\_\_AssetCodeRCB2     *Dictionary for AssetCodeRCB2 variable of RCB2 file*

---

**Description**

Dictionary for AssetCodeRCB2 variable of RCB2 file

**Usage**

RCB2\_\_AssetCodeRCB2

**Format**

RCB2\_\_AssetCodeRCB2:

A data frame with 17 rows and 2 columns:

**code** Integer code

**value** Character description

**Source**

D\_RCB2.TXT

---

RCB3\_\_DebtMaturityCode  
*Dictionary for DebtMaturityCode variable of RCB3 file*

---

**Description**

Dictionary for DebtMaturityCode variable of RCB3 file

**Usage**

RCB3\_\_DebtMaturityCode

**Format**

RCB3\_\_DebtMaturityCode:

A data frame with 10 rows and 2 columns:

**code** Integer code

**value** Character description

**Source**

D\_RCB3.TXT

---

RCB\_\_INV\_CODE                      *Dictionary for INV\_CODE variable of RCB file*

---

**Description**

Dictionary for INV\_CODE variable of RCB file

**Usage**

RCB\_\_INV\_CODE

**Format**

RCB\_\_INV\_CODE:

A data frame with 35 rows and 2 columns:

**code** Integer code

**value** Character description

**Source**

D\_RCB.TXT

---

RCF1\_\_LOANSTATUS                      *Dictionary for LOANSTATUS variable of RCF1 file*

---

**Description**

Dictionary for LOANSTATUS variable of RCF1 file

**Usage**

RCF1\_\_LOANSTATUS

**Format**

RCF1\_\_LOANSTATUS:

A data frame with 13 rows and 2 columns:

**code** Integer code

**value** Character description

**Source**

D\_RCF1.TXT

---

RCF\_\_LOANSTATUS      *Dictionary for LOANSTATUS variable of RCF file*

---

**Description**

Dictionary for LOANSTATUS variable of RCF file

**Usage**

RCF\_\_LOANSTATUS

**Format**

RCF\_\_LOANSTATUS:

A data frame with 6 rows and 2 columns:

**code** Integer code

**value** Character description

**Source**

D\_RCF.TXT

---

RCI2B\_\_DerivCode      *Dictionary for DerivCode variable of RCI2B file*

---

**Description**

Dictionary for DerivCode variable of RCI2B file

**Usage**

RCI2B\_\_DerivCode

**Format**

RCI2B\_\_DerivCode:

A data frame with 23 rows and 2 columns:

**code** Integer code

**value** Character description

**Source**

D\_RCI2B\_2018.TXT

---

RCI2C\_\_ExposureCode     *Dictionary for ExposureCode variable of RCI2C file*

---

**Description**

Dictionary for ExposureCode variable of RCI2C file

**Usage**

RCI2C\_\_ExposureCode

**Format**

RCI2C\_\_ExposureCode:

A data frame with 12 rows and 2 columns:

**code** Integer code

**value** Character description

**Source**

D\_RCI2C\_2018.TXT

---

RCI2D\_\_DerivRMCode     *Dictionary for DerivRMCode variable of RCI2D file*

---

**Description**

Dictionary for DerivRMCode variable of RCI2D file

**Usage**

RCI2D\_\_DerivRMCode

**Format**

RCI2D\_\_DerivRMCode:

A data frame with 11 rows and 2 columns:

**code** Integer code

**value** Character description

**Source**

D\_RCI2D\_2018.TXT

---

RCO\_\_ASSET\_CODE      *Dictionary for ASSET\_CODE variable of RCO file*

---

**Description**

Dictionary for ASSET\_CODE variable of RCO file

**Usage**

RCO\_\_ASSET\_CODE

**Format**

RCO\_\_ASSET\_CODE:

A data frame with 12 rows and 2 columns:

**code** Integer code

**value** Character description

**Source**

D\_RCO.TXT

---

RCR3\_\_RegCapCode      *Dictionary for RegCapCode variable of RCR3 file*

---

**Description**

Dictionary for RegCapCode variable of RCR3 file

**Usage**

RCR3\_\_RegCapCode

**Format**

RCR3\_\_RegCapCode:

A data frame with 15 rows and 2 columns:

**code** Integer code

**value** Character description

**Source**

D\_RCR3.TXT

---

RCR7__RegCapCode	<i>Dictionary for RegCapCode variable of RCR7 file</i>
------------------	--------------------------------------------------------

---

**Description**

Dictionary for RegCapCode variable of RCR7 file

**Usage**

RCR7\_\_RegCapCode

**Format**

RCR7\_\_RegCapCode:

A data frame with 29 rows and 2 columns:

**code** Integer code

**value** Character description

**Source**

D\_RCR7.TXT

---

read_data_file	<i>Read a data file based on metadata and codes dictionary</i>
----------------	----------------------------------------------------------------

---

**Description**

read\_data\_file() reads a data file and processes it based on the provided metadata and codes dictionary. The processing depends on the metadata scenario, which includes cases like "single", "single\_multiple", and "single\_multiple\_single". For certain scenarios, the function utilizes read.csv to infer column types without explicit specification.

**Usage**

```
read_data_file(file, metadata, dict)
```

**Arguments**

file	A character string specifying the path to the data file.
metadata	A list containing the scenario and variable information obtained from the metadata file using <a href="#">process_metadata_file</a> .
dict	A data frame containing codes dictionary information.

**Details**

`read_data_file()` reads the data file and applies necessary processing based on the metadata scenario. For scenarios like "single" and "single\_multiple", it uses `read.csv` for convenient type inference. For "single\_multiple\_single", it reads the file line by line, collapses every  $(N\_CODES + 2)$  lines, and then reads the collapsed lines using `read.table`.

**Value**

A tibble containing the processed data.

---

RID__CAP_CODE	<i>Dictionary for RegCapCode variable of RID file</i>
---------------	-------------------------------------------------------

---

**Description**

Dictionary for RegCapCode variable of RID file

**Usage**

RID\_\_CAP\_CODE

**Format**

RID\_\_CAP\_CODE:

A data frame with 12 rows and 2 columns:

**code** Integer code

**value** Character description

**Source**

D\_RID.TXT

---

RIE1__ACLCode	<i>Dictionary for ACLCode variable of RIE1 file</i>
---------------	-----------------------------------------------------

---

**Description**

Dictionary for ACLCode variable of RIE1 file

**Usage**

RIE1\_\_ACLCode



**Format**

RIE1\_\_ACLCode:

A data frame with 7 rows and 2 columns:

**code** Integer code

**value** Character description

**Source**

D\_RIE1.TXT

# Index

## \* datasets

- file\_metadata, 5
- RCB2\_\_AssetCodeRCB2, 10
- RCB3\_\_DebtMaturityCode, 10
- RCB\_\_INV\_CODE, 11
- RCF1\_\_LOANSTATUS, 11
- RCF\_\_LOANSTATUS, 12
- RCI2B\_\_DerivCode, 12
- RCI2C\_\_ExposureCode, 13
- RCI2D\_\_DerivRMCode, 13
- RCO\_\_ASSET\_CODE, 14
- RCR3\_\_RegCapCode, 14
- RCR7\_\_RegCapCode, 15
- RID\_\_CAP\_CODE, 16
- RIE1\_\_ACLCode, 16

compare\_files\_content, 2

compare\_metadata, 3

download\_data, 4

file\_metadata, 5

get\_codes\_dict, 6

process\_data, 7

process\_data\_file, 8

process\_metadata\_file, 8, 9, 15

  

- RCB2\_\_AssetCodeRCB2, 10
- RCB3\_\_DebtMaturityCode, 10
- RCB\_\_INV\_CODE, 11
- RCF1\_\_LOANSTATUS, 11
- RCF\_\_LOANSTATUS, 12
- RCI2B\_\_DerivCode, 12
- RCI2C\_\_ExposureCode, 13
- RCI2D\_\_DerivRMCode, 13
- RCO\_\_ASSET\_CODE, 14
- RCR3\_\_RegCapCode, 14
- RCR7\_\_RegCapCode, 15

read\_data\_file, 8, 15

RID\_\_CAP\_CODE, 16

RIE1\_\_ACLCode, 16