

# Package ‘edgar’

August 10, 2023

**Type** Package

**Title** Tool for the U.S. SEC EDGAR Retrieval and Parsing of Corporate Filings

**Version** 2.0.6

**Date** 2023-08-10

**Author** Gunratan Lonare <lonare.gunratan@gmail.com>, Bharat Patil  
<bharatspatil@gmail.com>

**Maintainer** Gunratan Lonare <lonare.gunratan@gmail.com>

**Depends** R (>= 3.5.0)

**Imports** R.utils, tm, XML, stringr, stringi, qdapRegex, httr

**Description** In the USA, companies file different forms with the U.S. Securities and Exchange Commission (SEC) through EDGAR (Electronic Data Gathering, Analysis, and Retrieval system). The EDGAR database automated system collects all the different necessary filings and makes it publicly available. This package facilitates retrieving, storing, searching, and parsing of all the available filings on the EDGAR server. It downloads filings from SEC server in bulk with a single query. Additionally, it provides various useful functions: extracts 8-K triggering events, extract ``Business (Item 1)" and ``Management's Discussion and Analysis(Item 7)" sections of annual statements, searches filings for desired keywords, provides sentiment measures, parses filing header information, and provides HTML view of SEC filings.

**License** GPL-2

**RoxygenNote** 7.2.3

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2023-08-10 13:20:02 UTC

## R topics documented:

get8KItems	2
getBusinDescr	3
getDailyMaster	4
getFilingHeader	5
getFilingInfo	6
getFilings	7
getFilingsHTML	8
getMasterIndex	9
getMgmtDisc	10
getSentiment	11
LMMasterDictionary	13
searchFilings	13
<b>Index</b>	<b>15</b>

---

get8KItems	<i>Retrieves Form 8-K event information</i>
------------	---

---

### Description

get8KItems retrieves Form 8-K event information of firms based on CIK numbers and filing year.

### Usage

```
get8KItems(cik.no, filing.year, useragent)
```

### Arguments

cik.no	vector of CIK(s) in integer format. Suppress leading zeroes from CIKs.
filing.year	vector of four digit numeric year
useragent	Should be in the form of "Your Name Contact@domain.com"

### Details

get8KItems function takes firm CIK(s) and filing year(s) as input parameters from a user and provides information on the Form 8-K triggering events along with the firm filing information. The function searches and imports existing downloaded 8-K filings in the current directory; otherwise it downloads them using [getFilings](#) function. It then reads the 8-K filings and parses them to extract events information. According to SEC EDGAR's guidelines a user also needs to declare user agent.

### Value

Function returns dataframe with Form 8-K events information along with CIK number, company name, date of filing, and accession number.

## Examples

```
## Not run:

output <- get8KItems(cik.no = 38079, filing.year = 2005, useragent)
## Returns 8-K event information for CIK '38079' filed in year 2005.

output <- get8KItems(cik.no = c(1000180,38079),
                    filing.year = c(2005, 2006), useragent)

## End(Not run)
```

---

getBusinDescr	<i>Retrieves business descriptions from annual statements</i>
---------------	---

---

## Description

getBusinessDescr retrieves business description section from annual statements based on CIK number(s) and filing year(s).

## Usage

```
getBusinDescr(cik.no, filing.year, useragent)
```

## Arguments

cik.no	vector of firm CIK(s) in integer format. Suppress leading zeroes from a CIK number. cik.no = 'ALL' considers all the CIKs.
filing.year	vector of four digit numeric year
useragent	Should be in the form of "Your Name Contact@domain.com"

## Details

getBusinDescr function takes firm CIK(s) and filing year(s) as input parameters from a user and provides "Item 1" section extracted from annual statements along with filing information. The function imports annual filings (10-K statements) downloaded via [getFilings](#) function; otherwise, it automates the downloading process if not already been downloaded. It then reads the downloaded statements, cleans HTML tags, and parse the contents. This function automatically creates a new directory with the name "Business descriptions text" in the current working directory and saves scrapped business description sections in this directory. It considers "10-K", "10-K405", "10KSB", and "10KSB40" form types as annual statements. According to SEC EDGAR's guidelines a user also needs to declare user agent.

## Value

Function saves scrapped business description section from annual filings in "Business descriptions text" directory created in the current working directory. The output dataframe contains filing information and parsing status. For a successful extraction of this section, 'extract.status' column returns 1, other return 0 for failed extraction.

## Examples

```
## Not run:

output <- getBusinDescr(cik.no = c(1000180, 38079), filing.year = 2005, useragent)
## saves scrapped "Item 1" section from 10-K filings for CIKs in
"Business descriptions text" directory present
in the working directory. Also, it provides filing information in
the output dataframe.

output <- getBusinDescr(cik.no = c(1000180, 38079),
                        filing.year = c(2005, 2006), useragent)

## End(Not run)
```

---

getDailyMaster	<i>Retrieves daily master index</i>
----------------	-------------------------------------

---

## Description

getDailyMaster retrieves daily master index from the U.S. SEC EDGAR server.

## Usage

```
getDailyMaster(input.date, useragent)
```

## Arguments

input.date	in character format 'mm/dd/YYYY'.
useragent	Should be in the form of "Your Name Contact@domain.com"

## Details

getDailyMaster function takes date as an input parameter from a user, and downloads master index for the date from the U.S. SEC EDGAR server <https://www.sec.gov/Archives/edgar/daily-index/>. It strips headers and converts this daily filing information into dataframe format. Function creates new directory 'Daily Indexes' into working directory to save these downloaded daily master index files in Rda format. According to SEC EDGAR's guidelines a user also needs to declare user agent.

## Value

Function returns filings information in a dataframe format.

## Examples

```
## Not run:  
  
output <- getDailyMaster('08/09/2016', useragent)  
  
## End(Not run)
```

---

getFilingHeader	<i>Scrape EDGAR filing header information</i>
-----------------	---

---

## Description

getFilingHeader Extract EDGAR filing header information

## Usage

```
getFilingHeader(cik.no, form.type, filing.year, useragent)
```

## Arguments

cik.no	vector of CIK(s) in integer format. Suppress leading zeroes from CIKs. cik.no = 'ALL' considers all the CIKs.
form.type	character vector containing form type to be downloaded. form.type = 'ALL' if need to download all forms.
filing.year	vector of four digit numeric year
useragent	Should be in the form of "Your Name Contact@domain.com"

## Details

getFilingHeader function takes CIK(s), form type(s), and year(s) as input parameters. The function first imports available downloaded filings in local working directory 'Edgar filings\_full text' created by [getFilings](#) function; otherwise, it automatically downloads the filings which are not already been downloaded. It then parses all the important header information from filings. The function returns a dataframe with filing and header information. According to SEC EDGAR's guidelines a user also needs to declare user agent.

## Value

Function returns dataframe containing CIK number, company name, date of filing, accession number, confirmed period of report, fiscal year end, Standard Industrial Classification (SIC) code, Internal Revenue Code (IRS) code, state of incorporation, business address, and mailing address. If a filing contains multiple filers then output will contain header information on all the filers in multiple rows.

**Examples**

```
## Not run:

header.df <- getFilingHeader(cik.no = c('1000180', '38079'),
                             form.type = '10-K', filing.year = 2006, useragent)

header.df <- getFilingHeader(cik.no = '38079', c('10-K', '10-Q'),
                             filing.year = c(2005, 2006), useragent)

## End(Not run)
```

---

<code>getFilingInfo</code>	<i>Retrieves filing information of a firm</i>
----------------------------	---

---

**Description**

`getFilingInfo` retrieves filing information of a firm based on its name or cik.

**Usage**

```
getFilingInfo(firm.identifier, filing.year, quarter, form.type, useragent)
```

**Arguments**

<code>firm.identifier</code>	CIK of a firm in integer format or full/partial name of a firm in character format. Suppress leading zeroes from CIKs.
<code>filing.year</code>	vector of integer containing filing years.
<code>quarter</code>	vector of one digit integer quarter number. By default, it is considered as all the quarters, <code>quarter = c(1, 2, 3, 4)</code> .
<code>form.type</code>	vector of form types in character format. By default, it is kept as all the available form types.
<code>useragent</code>	Should be in the form of "Your Name Contact@domain.com"

**Details**

`getFilingInfo` function takes firm identifier (name or cik), filing year(s), quarter(s), and form type as input parameters from a user and provides filing information for the firm. The function automatically downloads master index for the input year(s) and the quarter(s) using [getMasterIndex](#) function if it is not already been downloaded in the current working directory. By default, information of all the form types filed in all the quarters of the input year by the firm will be provided by this function. According to SEC EDGAR's guidelines a user also needs to declare user agent.

**Value**

Function returns dataframe with filing information.

**Examples**

```
## Not run:

info <- getFilingInfo('United Technologies', c(2005, 2006),
                    quarter = c(1,2), form.type = c('8-K','10-K'), useragent)
## Returns filing information on '8-K' and '10-K' filed by the firm
in quarter 1 and 2 of year 2005 and 2006.

info <- getFilingInfo(1067701, 2006, useragent)
## Returns all the filings information filed by the firm in all
the quarters of year 2006.

## End(Not run)
```

---

getFilings

*Retrieves EDGAR filings from SEC server*


---

**Description**

getFilings retrieves EDGAR filings for a specific CIKs, form-type, filing year and quarter of the filing.

**Usage**

```
getFilings(cik.no, form.type, filing.year, quarter, downl.permit, useragent)
```

**Arguments**

cik.no	vector of CIK number of firms in integer format. Suppress leading zeroes from CIKs. Keep cik.no = 'ALL' if needs to download for all CIKs.
form.type	character vector containing form type to be downloaded. form.type = 'ALL' if need to download all forms.
filing.year	vector of four digit numeric year
quarter	vector of one digit quarter integer number. By default, it is kept as c(1, 2, 3, 4).
downl.permit	"y" or "n". The default value of downl.permit is "n". It asks a user permission to download filings. This permission helps the user to decide in case if number of filings are large. Setting downl.permit = "y" will not ask for user permission to download filings.
useragent	Should be in the form of "Your Name Contact@domain.com"

**Details**

getFilings function takes CIKs, form type, filing year, and quarter of the filing as input. It creates new directory "Edgar filings\_full text" to store all downloaded filings. All the filings will be stored in the current working directory. Keep the same current working directory for further process. According to SEC EDGAR's guidelines a user also needs to declare user agent.

**Value**

Function downloads EDGAR filings and returns download status in dataframe format with CIK, company name, form type, date filed, accession number, and download status.

**Examples**

```
## Not run:

output <- getFilings(cik.no = c(1000180, 38079), c('10-K','10-Q'),
                    2006, quarter = c(1, 2, 3), downl.permit = "n", useragent)

## download '10-Q' and '10-K' filings filed by the firm with
## CIK = 1000180 in quarters 1,2, and 3 of the year 2006. These
## filings will be stored in the current working directory.

## End(Not run)
```

---

getFilingsHTML	<i>Get HTML view of EDGAR filings</i>
----------------	---------------------------------------

---

**Description**

getFilingsHTML retrieves complete EDGAR filings and store them in HTML format for view.

**Usage**

```
getFilingsHTML(cik.no, form.type, filing.year, quarter, useragent)
```

**Arguments**

cik.no	vector of CIK number of firms in integer format. Suppress leading zeroes from CIKs. Keep cik.no = 'ALL' if needs to download for all CIKs.
form.type	character vector containing form type to be downloaded. form.type = 'ALL' if need to download all forms.
filing.year	vector of four digit numeric year
quarter	vector of one digit quarter integer number. By default, it is kept as c(1 ,2, 3, 4).
useragent	Should be in the form of "Your Name Contact@domain.com"

**Details**

getFilingsHTML function takes CIK(s), form type(s), filing year(s), and quarter of the filing as input. The function imports edgar filings downloaded via [getFilings](#) function; otherwise, it downloads the filings which are not already been downloaded. It then reads the downloaded filings, scraps filing text excluding exhibits, and saves the filing contents in 'Edgar filings\_HTML view' directory in HTML format. The new directory 'Edgar filings\_HTML view' will be automatically created by this function. According to SEC EDGAR's guidelines a user also needs to declare user agent.



**Value**

Function saves EDGAR filings in HTML format and returns filing information in dataframe format.

**Examples**

```
## Not run:

output <- getFilingsHTML(cik.no = c(1000180, 38079), c('10-K','10-Q'),
                        2006, quarter = c(1, 2, 3), useragent)

## download '10-Q' and '10-K' filings filed by the firm with
## CIK = 1000180 in quarters 1,2, and 3 of the year 2006. These filings
## will be stored in the current working directory.

## End(Not run)
```

---

getMasterIndex	<i>Retrieves quarterly master index</i>
----------------	---

---

**Description**

getMasterIndex retrieves the quarterly master indexes from the U.S. SEC EDGAR server.

**Usage**

```
getMasterIndex(filing.year, useragent)
```

**Arguments**

filing.year	vector of integer containing filing years.
useragent	Should be in the form of "Your Name Contact@domain.com"

**Details**

getMasterIndex function takes filing year as an input parameter from a user, downloads quarterly master indexes from the US SEC server. <https://www.sec.gov/Archives/edgar/full-index/>. It then strips headers from the master index files, converts them into dataframe, and merges such quarterly dataframes into yearly dataframe, and stores them in Rda format. It has ability to download master indexes for multiple years based on the user input. This function creates a new directory 'Master Indexes' into current working directory to save these Rda Master Index. Please note, for all other functions in this package need to locate the same working directory to access these Rda master index files. According to SEC EDGAR's guidelines a user also needs to declare user agent.

**Value**

Function downloads quarterly master index files and stores them into the mentioned directory.

**Examples**

```
## Not run:

useragent <- "Your Name Contact@domain.com"

getMasterIndex(2006, useragent)
## Downloads quarterly master index files for 2006 and
stores into yearly 2006master.Rda file.

getMasterIndex(c(2006, 2008), useragent)
## Downloads quarterly master index files for 2006 and 2008, and
stores into 2006master.Rda and 2008master.Rda files.

## End(Not run)
```

---

```
getMgmtDisc
```

```
Retrieves management's discussion and analysis section
```

---

**Description**

getMgmtDisc retrieves "Item 7. Management's Discussion and Analysis of Financial Condition and Results of Operations" section of firms from annual statements based on CIK number and filing year.

**Usage**

```
getMgmtDisc(cik.no, filing.year, useragent)
```

**Arguments**

cik.no	vector of firm CIK numbers in integer format. Suppress leading zeroes from CIKs.
filing.year	vector of four digit numeric year
useragent	Should be in the form of "Your Name Contact@domain.com"

**Details**

getMgmtDisc function takes firm CIK(s) and filing year(s) as input parameters from a user and provides "Item 7" section extracted from annual statements along with filing information. The function imports annual filings downloaded via [getFilings](#) function; otherwise, it downloads the filings which are not already been downloaded. It then reads, cleans, and parse the required section from the filings. It creates a new directory with the name "MD&A section text" in the current working directory to save scrapped "Item 7" sections in text format. It considers "10-K", "10-K405", "10KSB", and "10KSB40" form types as annual statements. According to SEC EDGAR's guidelines a user also needs to declare user agent.

**Value**

Function saves scrapped "Item 7" section from annual filings in "MD&A section text" directory present in the working directory. The output dataframe contains information on CIK number, company name, date of filing, and accession number. For a successful extraction of M&A section, 'extract.status' column returns 1, other return 0 for failed extraction.

**Examples**

```
## Not run:

output <- getMgmtDisc(cik.no = c(1000180, 38079), filing.year = 2005, useragent)

## saves scrapped "Item 7" section from 10-K filings for CIKs in
"MD&A section text" directory present in the working directory.
Also, it provides filing information in the output dataframe.

output <- getMgmtDisc(cik.no = c(1000180, 38079),
                     filing.year = c(2005, 2006), useragent)

## End(Not run)
```

---

getSentiment	<i>Provides sentiment measures of EDGAR filings</i>
--------------	---

---

**Description**

getSentiment computes sentiment measures of EDGAR filings

**Usage**

```
getSentiment(cik.no, form.type, filing.year, useragent)
```

**Arguments**

cik.no	vector of CIK number of firms in integer format. Suppress leading zeroes from CIKs. Keep cik.no = 'ALL' if needs to download for all CIKs.
form.type	character vector containing form type to be downloaded. form.type = 'ALL' if need to download all forms.
filing.year	vector of four digit numeric year
useragent	Should be in the form of "Your Name Contact@domain.com"

**Details**

getSentiment function takes CIK(s), form type(s), and year(s) as input parameters. The function first imports available downloaded filings in the local working directory 'Edgar filings\_full text' created by [getFilings](#) function; otherwise, it automatically downloads the filings which are not already been downloaded. It then reads, cleans, and computes sentiment measures for these filings. The function returns a dataframe with filing information and sentiment measures. According to SEC EDGAR's guidelines a user also needs to declare user agent.

**Value**

Function returns dataframe containing CIK number, company name, date of filing, accession number, and various sentiment measures. This function takes the help of Loughran-McDonald (L&M) sentiment dictionaries (<https://sraf.nd.edu/loughranmcdonald-master-dictionary/>) to compute sentiment measures of a EDGAR filing. Following are the definitions of the text characteristics and the sentiment measures:

file.size = The filing size of a complete filing on the EDGAR server in kilobyte (KB).

word.count = The total number of words in a filing text, excluding HTML tags and exhibits text.

unique.word.count = The total number of unique words in a filing text, excluding HTML tags and exhibits text.

stopword.count = The total number of stop words in a filing text, excluding exhibits text.

char.count = The total number of characters in a filing text, excluding HTML tags and exhibits text.

complex.word.count = The total number of complex words in the filing text. When vowels (a, e, i, o, u) occur more than three times in a word, then that word is identified as a complex word.

lm.dictionary.count = The number of words in the filing text that occur in the Loughran-McDonald (LM) master dictionary.

lm.negative.count = The number of LM financial-negative words in the document.

lm.positive.count = The number of LM financial-positive words in the document.

lm.strong.modal.count = The number of LM financial-strong modal words in the document.

lm.moderate.modal.count = The number of LM financial-moderate Modal words in the document.

lm.weak.modal.count = The number of LM financial-weak modal words in the document.

lm.uncertainty.count = The number of LM financial-uncertainty words in the document.

lm.litigious.count = The number of LM financial-litigious words in the document.

hv.negative.count = The number of words in the document that occur in the 'Harvard General Inquirer' Negative word list, as defined by LM.

**Examples**

```
## Not run:

senti.df <- getSentiment(cik.no = c('1000180', '38079'),
                        form.type = '10-K', filing.year = 2006, useragent)

## Returns dataframe with sentiment measures of firms with CIKs
1000180 and 38079 filed in year 2006 for form type '10-K'.

senti.df <- getSentiment(cik.no = '38079', form.type = c('10-K', '10-Q'),
                        filing.year = c(2005, 2006), useragent)

## End(Not run)
```

---

LMMasterDictionary      *Loughran and McDonald Sentiment Master Dictionary*

---

**Description**

The data contains sentiments word lists.

**Details**

The sentiment categories are: negative, positive, uncertainty, litigious, modal, and Harvard IV. Modal words are flagged as 1, 2 or 3, with 1 = Strong Modal, 2 = Moderate Modal, and 3 = Weak Modal.

**Source**

Website: <https://sraf.nd.edu/loughranmcdonald-master-dictionary/>

**References**

Tim Loughran and Bill McDonald, 2011, When is a Liability not a Liability? Textual Analysis, Dictionaries, and 10-Ks, *Journal of Finance*, 66:1, 35-65.

Andriy Bodnaruk, Tim Loughran and Bill McDonald, 2015, Using 10-K Text to Gauge Financial Constraints, *Journal of Financial and Quantitative Analysis*, 50:4, 1-24.

Tim Loughran and Bill McDonald, 2016, Textual Analysis in Accounting and Finance: A Survey, *Journal of Accounting Research*, 54:4,1187-1230.

---

searchFilings      *Search EDGAR filings for specific keywords*

---

**Description**

searchFilings Search EDGAR filings for specific keywords

**Usage**

searchFilings(cik.no, form.type, filing.year, word.list, useragent)

**Arguments**

cik.no	vector of CIK number of firms in integer format. Suppress leading zeroes from CIKs. Keep cik.no = 'ALL' if needs to download for all CIK's.
form.type	character vector containing form type to be downloaded. form.type = 'ALL' if need to download all forms.
filing.year	vector of four digit numeric year
word.list	vector of words to search in the filing
useragent	Should be in the form of "Your Name Contact@domain.com"

## Details

searchFilings function takes search keyword vector, CIK(s), form type(s), and year(s) as input parameters. The function first imports available downloaded filings in the local working directory 'Edgar filings\_full text' created by [getFilings](#) function; otherwise, it automatically downloads the filings which are not already been downloaded. It then reads the filings and searches for the input keywords. The function returns a dataframe with filing information and the number of keyword hits. Additionally, it saves the search information with surrounding content of search keywords in HTML format in the new directory "Keyword search results". These HTML view of search results would help the user to analyze the search strategy and identify false positive hits. According to SEC EDGAR's guidelines a user also needs to declare user agent.

## Value

Function returns dataframe containing filing information and the number of word hits based on the input phrases. Additionally, this function saves search information with surrounding content of search keywords in HTML file in directory "Keyword search results".

## Examples

```
## Not run:

word.list = c('derivative','hedging','currency forwards','currency futures')
output <- searchFilings(cik.no = c('1000180', '38079'),
                      form.type = c("10-K", "10-K405", "10KSB", "10KSB40"),
                      filing.year = c(2005, 2006), word.list, useragent)

## End(Not run)
```

# Index

## \* data

LMMasterDictionary, [13](#)

[get8KItems, 2](#)

[getBusinDescr, 3](#)

[getDailyMaster, 4](#)

[getFilingHeader, 5](#)

[getFilingInfo, 6](#)

[getFilings, 2, 3, 5, 7, 8, 10, 11, 14](#)

[getFilingsHTML, 8](#)

[getMasterIndex, 6, 9](#)

[getMgmtDisc, 10](#)

[getSentiment, 11](#)

[LMMasterDictionary, 13](#)

[searchFilings, 13](#)