

# Package ‘parseRPDR’

May 9, 2026

**Type** Package

**Title** Parse and Manipulate Research Patient Data Registry ('RPDR')  
Text Queries

**Version** 1.1.2

**Date** 2025-01-19

**Maintainer** Marton Kolossvary <mkolossvary@ngh.harvard.edu>

**Description** Functions to load Research Patient Data Registry ('RPDR') text queries from Partners Healthcare institutions into R.  
The package also provides helper functions to manipulate data and execute common procedures such as finding the closest radiological exams considering a given timepoint, or creating a DICOM header database from the downloaded images. All functionalities are parallelized for fast and efficient analyses.

**License** AGPL (>= 3)

**Depends** R (>= 4.0)

**Imports** data.table (>= 1.14.1), stringr (>= 1.4.0), readr (>= 1.4.0),  
parallelly (>= 1.36.0), foreach (>= 1.5.1), future (>= 1.33.1),  
doFuture (>= 1.0.1), progressr (>= 0.14.0)

**RoxygenNote** 7.3.2

**NeedsCompilation** no

**Suggests** testthat (>= 3.0.0), reticulate (>= 1.20), knitr, rmarkdown,  
covr

**Encoding** UTF-8

**URL** <https://github.com/martonkolossvary/parseRPDR>

**BugReports** <https://github.com/martonkolossvary/parseRPDR/issues>

**Config/testthat/edition** 3

**Config/testthat/parallel** false

**Config/testthat/start-first** load\_all\_data, create\_img\_db, find\_exam,  
load\_\*, convert\_\*

**Author** Marton Kolossvary [aut, cre]

**Repository** CRAN

**Date/Publication** 2025-01-19 18:10:02 UTC

## Contents

all_ids_mi2b2 . . . . .	3
convert_dia . . . . .	3
convert_enc . . . . .	5
convert_lab . . . . .	6
convert_med . . . . .	7
convert_notes . . . . .	9
convert_phy . . . . .	11
convert_prc . . . . .	12
convert_rfv . . . . .	13
create_img_db . . . . .	15
export_notes . . . . .	16
find_exam . . . . .	17
load_all . . . . .	20
load_all_data . . . . .	22
load_bib . . . . .	23
load_con . . . . .	25
load_dem . . . . .	28
load_dem_old . . . . .	30
load_dia . . . . .	32
load_enc . . . . .	34
load_lab . . . . .	37
load_lno . . . . .	39
load_mcm . . . . .	41
load_med . . . . .	43
load_mic . . . . .	45
load_mrn . . . . .	47
load_notes . . . . .	49
load_phy . . . . .	51
load_prc . . . . .	53
load_prv . . . . .	55
load_ptd . . . . .	57
load_rdt . . . . .	59
load_rfv . . . . .	61
load_trn . . . . .	63
pretty_mrn . . . . .	65
pretty_numbers . . . . .	66
pretty_text . . . . .	66
remove_column . . . . .	67

---

all_ids_mi2b2	<i>Legacy function to create a vector of all possible IDs for mi2b2 workbench</i>
---------------	---

---

**Description**

Legacy function to gather all possible MGH and BWH IDs from mrn.txt and con.txt input sources to provide a vector of all possible MGH or BWH IDs to be used as a data request for mi2b2 workbench.

**Usage**

```
all_ids_mi2b2(type = "MGH", d_mrn, d_con)
```

**Arguments**

type	string, either "MGH" or "BWH" specifying which IDs to use.
d_mrn	data.table, parsed mrn dataset using the <i>load_mrn</i> function.
d_con	data.table, parsed con dataset using the <i>load_con</i> function.

**Value**

vector, with all MGH or BWH IDs that occur in the con and mrn datasources for all patients. Previously this was required to for mi2b2 workbenches allowing access to all possible images of the patients, even if the MGH or BWH changed over time.

**Examples**

```
## Not run:
all_MGH_mrn <- all_ids_mi2b2(type = "MGH", d_mrn = data_mrn, d_con = data_con)

## End(Not run)
```

---

convert_dia	<i>Searches diagnosis columns for given diseases.</i>
-------------	---

---

**Description**

Analyzes diagnosis data loaded using *load\_dia*. Searches diagnosis columns for a specified set of diseases. By default, the data.table is returned with new columns corresponding to boolean values, whether given group of diagnoses are present among the diagnoses. If *collapse* is given, then the information is aggregated based-on the *collapse* column and the earliest of latest time of the given diagnosis is provided.

**Usage**

```

convert_dia(
  d,
  code = "dia_code",
  code_type = "dia_code_type",
  codes_to_find = NULL,
  collapse = NULL,
  code_time = "time_dia",
  aggr_type = "earliest",
  nThread = parallel::detectCores() - 1
)

```

**Arguments**

d	data.table, database containing diagnosis information data loaded using the <i>load_dia</i> function.
code	string, column name of the diagnosis code column. Defaults to <i>dia_code</i> .
code_type	string, column name of the code_type column. Defaults to <i>dia_code_type</i> .
codes_to_find	list, a list of string arrays corresponding to sets of code types and codes separated by :, i.e.: "ICD9:250.00". The function searches for the given disease code type and code pair and adds new boolean columns with the name of each list element. These columns are indicators whether any of the disease code type and code pair occurs in the set of codes.
collapse	string, a column name on which to collapse the data.table. Used in case we wish to assess whether given disease codes are present within all the same instances of <i>collapse</i> . See vignette for details.
code_time	string, column name of the time column. Defaults to <i>time_dia</i> . Used in case collapse is present to provide the earliest or latest instance of diagnosing the given disease.
aggr_type	string, if multiple diagnoses are present within the same case of <i>collapse</i> , which timepoint to return. Supported are: "earliest" or "latest". Defaults to <i>earliest</i> .
nThread	integer, number of threads to use for parallelization. If it is set to 1, then no parallel backends are created and the function is executed sequentially.

**Value**

data.table, with indicator columns whether the any of the given diagnoses are reported. If *collapse* is present, then only unique ID and the summary columns are returned.

**Examples**

```

## Not run:
#Search for Hypertension and Stroke ICD codes
diseases <- list(HT = c("ICD10:I10"), Stroke = c("ICD9:434.91", "ICD9:I63.50"))
data_dia_parse <- convert_dia(d = data_dia, codes_to_find = diseases, nThread = 2)

#Search for Hypertension and Stroke ICD codes and summarize per patient providing earliest time

```

```
diseases <- list(HT = c("ICD10:I10"), Stroke = c("ICD9:434.91", "ICD9:I63.50"))
data_dia_disease <- convert_dia(d = data_dia, codes_to_find = diseases, nThread = 2,
collapse = "ID_MERGE", aggr_type = "earliest")

## End(Not run)
```

---

 convert\_enc

*Searches columns for given diseases defined by ICD codes.*


---

## Description

Analyzes encounter data loaded using *load\_enc*. Converts columns with ICD codes and text to simple ICD codes. If requested, the *data.table* is returned with new columns corresponding to boolean values, whether given group of diagnoses are present in the given columns. If *collapse* is given, then the information is aggregated based-on the *collapse* column and the earliest of latest time of the given diagnosis is provided.

## Usage

```
convert_enc(
  d,
  code = c("enc_diag_admit", "enc_diag_princ", paste0("enc_diag_", 1:10)),
  keep = FALSE,
  codes_to_find = NULL,
  collapse = NULL,
  code_time = "time_enc_admit",
  aggr_type = "earliest",
  nThread = parallel::detectCores() - 1
)
```

## Arguments

- |               |  |
|---------------|--|
| d             | data.table, database containing encounter information data loaded using the <i>load_enc</i> function.  |
| code          | string vector, an array of column names to convert to simple ICD codes. The new column names will be the old one with <i>ICD_</i> added to the beginning of it.  |
| keep          | boolean, whether to keep original columns that were converted. Defaults to <i>FALSE</i> .  |
| codes_to_find | list, a list of arrays corresponding to sets of ICD codes. The function searches the columns in code and new boolean columns with the name of each list element will be created. These columns are indicators whether the given disease is present in the set of ICD codes or not. |
| collapse      | string, a column name on which to collapse the <i>data.table</i> . Used in case we wish to assess whether given diagnoses are present within all the same instances of <i>collapse</i> . See vignette for details.   |

code_time	string, column name of the time column. Defaults to <i>time_enc_admit</i> . Used in case collapse is present to provide the earliest or latest instance of diagnosing the given disease.
aggr_type	string, if multiple diagnoses are present within the same case of <i>collapse</i> , which timepoint to return. Supported are: "earliest" or "latest". Defaults to <i>earliest</i> .
nThread	integer, number of threads to use for parallelization. If it is set to 1, then no parallel backends are created and the function is executed sequentially.

### Value

data.table, with formatted ICD code columns and possibly indicator columns if provided. If *collapse* is present, then only unique ID and the summary columns are returned.

### Examples

```
## Not run:
#Parse encounter ICD columns and keep original ones as well
data_enc_parse <- convert_enc(d = data_enc, keep = TRUE, nThread = 2)

#Parse encounter ICD columns and discard original ones,
#and create indicator variable for the following diseases
diseases <- list(HT = c("I10"), Stroke = c("434.91", "I63.50"))
data_enc_disease <- convert_enc(d = data_enc, keep = FALSE,
codes_to_find = diseases, nThread = 2)

#Parse encounter ICD columns and discard original ones
#and create indicator variables for the following diseases and summarize per patient,
#whether there are any encounters where the given diseases were registered
diseases <- list(HT = c("I10"), Stroke = c("434.91", "I63.50"))
data_enc_disease <- convert_enc(d = data_enc, keep = FALSE,
codes_to_find = diseases, nThread = 2, collapse = "ID_MERGE")

## End(Not run)
```

---

convert\_lab

*Converts lab results to normal/abnormal based-on reference values.*

---

### Description

Analyzes laboratory data loaded using *load\_lab*. Converts laboratory results to values without ">" or "<" by creating a column where these characters are removed. Furthermore, adds two indicator columns where based-on the reference ranges or the Abnormal\_Flag column in RPDR (lab\_result\_abn using *load\_lab*), the value is considered normal or abnormal.

**Usage**

```
convert_lab(
  d,
  code_results = "lab_result",
  code_reference = "lab_result_range",
  code_flag = "lab_result_abn"
)
```

**Arguments**

**d** data.table, database containing laboratory results data loaded using the *load\_lab* function.

**code\_results** string vector, column name containing the results. Defaults to: *"lab\_result"*.

**code\_reference** string vector, column name containing the reference ranges. Defaults to: *"lab\_result\_range"*.

**code\_flag** string vector, column name containing the abnormal flags. Defaults to: *"lab\_result\_abn"*.

**Value**

data.table, with three additional columns: *"lab\_result\_pretty"* containing numerical results. In case of ">" or "<" notation, the numeric value is returned, as we only have information that it is at least as much or not larger than a given value. The other column: *"lab\_result\_abn\_pretty"* can take values: NORMAL/ABNORMAL, depending on whether the value is within the reference range. Please be aware that there can be very different representations of values, and in some cases this will result in misclassification of values. The third column: *"lab\_result\_abn\_flag\_pretty"* gives abnormal if the original Abnormal\_Flag column contains any information. Borderline values are considered NORMAL.

**Examples**

```
## Not run:
#Convert loaded lab results
data_lab_pretty <- convert_lab(d = data_lab)
data_lab_pretty[, c("lab_result", "lab_result_pretty", "lab_result_range",
"lab_result_abn_pretty", "lab_result_abn_flag_pretty")]

## End(Not run)
```

---

convert_med	<i>Adds boolean columns corresponding to a group of medications whether it is present in the given row.</i>
-------------	---

---

**Description**

Analyzes medication data loaded using *load\_med*. By default, the data.table is returned with new columns corresponding to boolean values, whether given group of medications are present. If *collapse* is given, then the information is aggregated based-on the *collapse* column and the earliest of latest time of the given medication is provided.

**Usage**

```

convert_med(
  d,
  code = "med",
  codes_to_find = NULL,
  collapse = NULL,
  code_time = "time_med",
  aggr_type = "earliest",
  nThread = parallel::detectCores() - 1
)

```

**Arguments**

d	data.table, database containing medication data loaded using the <i>load_med</i> function.
code	string, column name of the medication column. Defaults to <i>med</i> .
codes_to_find	list, a list of arrays corresponding to sets of medication names. New boolean columns with the name of each list element will be created. These columns are indicators whether the given medication is present in the set of medication names or not.
collapse	string, a column name on which to collapse the data.table. Used in case we wish to assess whether given medications are present within all the same instances of <i>collapse</i> . See vignette for details.
code_time	string, column name of the time column. Defaults to <i>time_med</i> . Used in case collapse is present to provide the earliest or latest instance of diagnosing the given disease.
aggr_type	string, if multiple occurrences of the medications are present within the same case of <i>collapse</i> , which timepoint to return. Supported are: "earliest" or "latest". Defaults to <i>earliest</i> .
nThread	integer, number of threads to use for parallelization. If it is set to 1, then no parallel backends are created and the function is executed sequentially.

**Value**

data.table, with indicator columns whether given group of codes\_to\_find is present or not. If *collapse* is present, then only unique ID and the summary columns are returned.

**Examples**

```

## Not run:
#Define medication group and add an indicator column whether
#the given medication group was administered
meds <- list(statin = c("Simvastatin", "Atorvastatin"),
            NSAID = c("Acetaminophen", "Paracetamol"))

data_med_indic <- convert_med(d = data_med, codes_to_find = meds, nThread = 1)

```

```
#Summarize per patient if they ever had the given medication groups registered
data_med_indic_any <- convert_med(d = data_med,
codes_to_find = meds, collapse = "ID_MERGE", nThread = 2)

## End(Not run)
```

---

 convert\_notes

*Extracts information from notes free text.*


---

## Description

Analyzes notes loaded using *load\_notes* or *load\_Ino*. Extracts information from the free text present in *abc\_rep\_txt*, where *abc* stands for the three letter abbreviation of the given type of note. An array of string is provided using the *anchors* argument. The function will return as many columns as there are anchor points. Each column will contain the text between the given anchor point and the next following anchor point. This way the free text report is split into corresponding smaller texts. By default, these are the common standard elements of given note types. Here are provided potential anchor points for the given types of notes:

- Cardiology:** c("Report Number:", "Report Status:", "Type:", "Date:", "Ordering Provider:", "SYS-TOLIC BLOOD PRESSURE", "DIASTOLIC BLOOD PRESSURE", "VENTRICULAR RATE EKG/MIN", "ATRIAL RATE", "PR INTERVAL", "QRS DURATION", "QT INTERVAL", "QTC INTERVAL", "P AXIS", "R AXIS", "T WAVE AXIS", "LOC", "DX:", "REF:", "Electronically Signed", "report\_end")
- Discharge:** c("\*\*\*\*This text report", "Patient Information", "Physician Discharge Summary", "Surgeries this Admission", "Items for Post-Hospitalization Follow-Up:", "Pending Results", "Hospital Course", "ED Course:", "Diagnosis", "Prescriptions prior to admission", "Family History:", "Physical Exam on Admission:", "Discharge Exam", "report\_end")
- Endoscopy:** c("NAME:", "DATE:", "Patient Information", "report\_end")
- History & Physical:** c("\*\*\*This text report", "Patient Information", "H&P by", "Author:", "Service:", "Author Type:", "Filed:", "Note Time:", "Status:", "Editor:", "report\_end")
- Operative:** c("NAME:", "UNIT NO:", "DATE:", "SURGEON:", "ASST:", "PREOPERATIVE DIAGNOSIS:", "POSTOPERATIVE DIAGNOSIS:", "NAME OF OPERATION:", "ANESTHESIA:", "INDICATIONS", "OPERATIVE FINDINGS:", "DESCRIPTION OF PROCEDURE:", "Electronically Signed", "report\_end")
- Pathology:** c("Accession Number:", "Report Status:", "Type:", "Report:", "CASE:", "PATIENT:", "Date", "Source Care Unit:", "Path Subspecialty Service:", "Results To:", "Signed Out by:", "CLINICAL DATA:", "FINAL DIAGNOSIS:", "GROSS DESCRIPTION:", "report\_end")
- Progress:** c("\*\*\*\*This text report", "Patient Information", "History", "Overview", "Progress Notes", "Medications", "Relevant Orders", "Level of Service", "report\_end")
- Pulmonary:** c("The Pulmonary document", "Name:", "Unit #:", "Date:", "Location:", "Smoking Status:", "Pack Years:", "SPIROMETRY:", "LUNG VOLUMES:", "DIFFUSION:", "PLETHYSMOGRAPHY:" "Pulmonary Function Test Interpretation", "Spirometry", "report\_end")
- Radiology:** c("Exam Code", "Ordering Provider", "HISTORY", "Associated Reports", "Report Below", "REASON", "REPORT", "TECHNIQUE", "COMPARISON", "FINDINGS", "IMPRESSION", "RECOMMENDATION", "SIGNATURES", "report\_end")

**Visit:** `c("***This text report", "Reason for Visit", "Reason for Visit", "Vital Signs", "Chief Complaint", "History", "Overview", "Medications", "Relevant Orders", "Level of Service", "report_end"`

**LMR:** `c("Subject", "Patient Name:", "Reason for visit", "report_end"`

However, these may be modified and extended to include sections of interest, i.e. if a given score is reported in a standard fashion, then adding this phrase (i.e. "CAD-RADS") would create a column where the text following this statement is returned. After this the resulting columns can be easily cleaned up if needed. Be aware to always include "report\_end" in the anchors array, to provide the function of the last occurring statement in the report.

### Usage

```
convert_notes(
  d,
  code = NULL,
  anchors = NULL,
  nThread = parallel::detectCores() - 1
)
```

### Arguments

<code>d</code>	data.table, database containing notes loaded using the <i>load_notes</i> function.
<code>code</code>	string vector, column name containing the results, which should be "abc_rep_txt", where abc stands for the three letter abbreviation of the given type of note.
<code>anchors</code>	string array, elements to search for in the text report.
<code>nThread</code>	integer, number of threads to use for parallelization. If it is set to 1, then no parallel backends are created and the function is executed sequentially.

### Value

data.table, with new columns corresponding to elements in *anchors*.

### Examples

```
## Not run:
#Create columns with specific parts of the radiological report defined by anchors
data_rad_parsed <- convert_notes(d = data_rad, code = "rad_rep_txt",
  anchors = c("Exam Code", "Ordering Provider", "HISTORY", "Associated Reports",
  "Report Below", "REASON", "REPORT", "TECHNIQUE", "COMPARISON", "FINDINGS",
  "IMPRESSION", "RECOMMENDATION", "SIGNATURES", "report_end"), nThread = 2)

## End(Not run)
```

---

convert_phy	<i>Searches health history data for given codes</i>
-------------	---

---

### Description

Analyzes health history data loaded using *load\_phy*. Searches health history columns for a specified set of codes. By default, the `data.table` is returned with new columns corresponding to boolean values, whether given group of health history data are present within the respective columns. If *collapse* is given, then the information is aggregated based-on the *collapse* column and the earliest of latest time of the given diagnosis is provided.

### Usage

```
convert_phy(
  d,
  code = "phy_code",
  code_type = "phy_code_type",
  codes_to_find = NULL,
  collapse = NULL,
  code_time = "time_phy",
  aggr_type = "earliest",
  nThread = parallel::detectCores() - 1
)
```

### Arguments

<code>d</code>	<code>data.table</code> , database containing health history information data loaded using the <i>load_phy</i> function.
<code>code</code>	string, column name of the diagnosis code column. Defaults to <i>phy_code</i> .
<code>code_type</code>	string, column name of the <code>code_type</code> column. Defaults to <i>phy_code_type</i> .
<code>codes_to_find</code>	list, a list of string arrays corresponding to sets of code types and codes separated by <code>:</code> , i.e.: "LMR:3688". The function searches for the given health history code type and code pair and adds new boolean columns with the name of each list element. These columns are indicators whether any of the health history code type and code pair occurs in the set of codes.
<code>collapse</code>	string, a column name on which to collapse the <code>data.table</code> . Used in case we wish to assess whether multiple health history codes are present within all the same instances of <i>collapse</i> . See vignette for details.
<code>code_time</code>	string, column name of the time column. Defaults to <i>time_phy</i> . Used in case <i>collapse</i> is present to provide the earliest or latest instance of health history information.
<code>aggr_type</code>	string, if multiple health histories are present within the same case of <i>collapse</i> , which timepoint to return. Supported are: "earliest" or "latest". Defaults to <i>earliest</i> .
<code>nThread</code>	integer, number of threads to use for parallelization. If it is set to 1, then no parallel backends are created and the function is executed sequentially.

**Value**

data.table, with indicator columns whether the any of the given health histories are reported. If *collapse* is present, then only unique ID and the summary columns are returned.

**Examples**

```
## Not run:
#Search for Height and Weight codes
anthropometrics <- list(Weight = c("LMR:3688", "EPIC:WGT"), Height = c("LMR:3771", "EPIC:HGT"))
data_phy_parse <- convert_phy(d = data_phy, codes_to_find = anthropometrics, nThread = 2)

#Search for for Height and Weight codes and summarize per patient providing earliest time
anthropometrics <- list(Weight = c("LMR:3688", "EPIC:WGT"), Height = c("LMR:3771", "EPIC:HGT"))
data_phy_parse <- convert_phy(d = data_phy, codes_to_find = anthropometrics, nThread = 2,
collapse = "ID_MERGE", aggr_type = "earliest")

## End(Not run)
```

---

 convert\_prc

*Searches procedures columns for given procedures.*


---

**Description**

Analyzes procedure data loaded using *load\_prc*. Searches procedures columns for a specified set of procedures. By default, the data.table is returned with new columns corresponding to boolean values, whether given group of procedures are present in the given procedure. If *collapse* is given, then the information is aggregated based-on the *collapse* column and the earliest of latest time of the given procedure is provided.

**Usage**

```
convert_prc(
  d,
  code = "prc_code",
  code_type = "prc_code_type",
  codes_to_find = NULL,
  collapse = NULL,
  code_time = "time_prc",
  aggr_type = "earliest",
  nThread = parallel::detectCores() - 1
)
```

**Arguments**

d	data.table, database containing procedure information data loaded using the <i>load_prc</i> function.
code	string, column name of the procedure code column. Defaults to <i>prc_code</i> .

code_type	string, column name of the code_type column. Defaults to <i>prc_code_type</i> .
codes_to_find	list, a list of string arrays corresponding to sets of code types and codes separated by :, i.e.: "CPT:00104". The function searches for the given procedure code type and code pair and adds new boolean columns with the name of each list element. These columns are indicators whether any of the procedure code type and code pair occurs in the set of codes.
collapse	string, a column name on which to collapse the data.table. Used in case we wish to assess multiple procedure codes are present within all the same instances of <i>collapse</i> . See vignette for details.
code_time	string, column name of the time column. Defaults to <i>time_prc</i> . Used in case collapse is present to provide the earliest or latest instance of the given procedure.
aggr_type	string, if multiple procedures are present within the same case of <i>collapse</i> , which timepoint to return. Supported are: "earliest" or "latest". Defaults to <i>earliest</i> .
nThread	integer, number of threads to use for parallelization. If it is set to 1, then no parallel backends are created and the function is executed sequentially.

### Value

data.table, with indicator columns whether the any of the given procedures are reported. If *collapse* is present, then only unique ID and the summary columns are returned.

### Examples

```
## Not run:
#Search for Anesthesia CPT codes
procedures <- list(Anesthesia = c("CTP:00410", "CPT:00104"))
data_prc_parse <- convert_prc(d = data_prc, codes_to_find = procedures, nThread = 2)

#Search for Anesthesia CPT codes
procedures <- list(Anesthesia = c("CTP:00410", "CPT:00104"))
data_prc_procedures <- convert_prc(d = data_prc, codes_to_find = procedures,
nThread = 2, collapse = "ID_MERGE", aggr_type = "earliest")

## End(Not run)
```

---

convert\_rfv

*Searches columns for given reason for visit defined by ERFV codes.*

---

### Description

Analyzes reason for visit data loaded using *load\_rfv*. If requested, the data.table is returned with new columns corresponding to boolean values, whether given group of ERFV are present in the given columns. If *collapse* is given, then the information is aggregated based-on the *collapse* column and the earliest of latest time of the given reason for visit is provided.

**Usage**

```

convert_rfv(
  d,
  code = "rfv_concept_id",
  codes_to_find = NULL,
  collapse = NULL,
  code_time = "time_rfv_start",
  aggr_type = "earliest",
  nThread = parallel::detectCores() - 1
)

```

**Arguments**

d	data.table, database containing reason for visit information data loaded using the <i>load_rfv</i> function.
code	string vector, an array of column names to search.
codes_to_find	list, a list of arrays corresponding to sets of ERFV codes. The function searches the columns in code and the name of each list element will be created. These columns are indicators whether the given disease is present in the set of ERFV codes or not.
collapse	string, a column name on which to collapse the data.table. Used in case we wish to assess whether multiple ERFV are present within the same instances of <i>collapse</i> . See vignette for details.
code_time	string, column name of the time column. Defaults to <i>time_rfv_start</i> . Used in case collapse is present to provide the earliest or latest instance of reason for visit.
aggr_type	string, if multiple reason for visits are present within the same case of <i>collapse</i> , which timepoint to return. Supported are: "earliest" or "latest". Defaults to <i>earliest</i> .
nThread	integer, number of threads to use for parallelization. If it is set to 1, then no parallel backends are created and the function is executed sequentially.

**Value**

data.table, with indicator columns if provided. If *collapse* is present, then only unique ID and the summary columns are returned.

**Examples**

```

## Not run:
#Parse reason for visit columns
#and create indicator variables for the following reasons and summarize per patient,
#whether there are any encounters where the given reasons were registered
reasons <- list(Pain = c("ERFV:160357", "ERFV:140012"), Visit = c("ERFV:501"))
data_rfv_disease <- convert_rfv(d = data_rfv, keep = FALSE,
codes_to_find = reasons, nThread = 2, collapse = "ID_MERGE")

## End(Not run)

```

---

create_img_db	<i>Create a database of DICOM headers.</i>
---------------	--

---

## Description

The function creates a database of DICOM headers present in a folder structure. Each series should be in its own folder, but they can be in a nested folder structure. Files where there are also folder present next to them at the same level will not be parsed. That is the folder structure needs to comply with the DICOM standard. Be aware that the function requires `python` and `pydicom` to be installed! The function cycles through all folders present in the provided path and recursively goes through them, every subfolder, and extracts the DICOM header information from the files using the `dcmread` function of the `pydicom` package. The extension of the files can be provided by the `ext` argument, as DICOM files may have different extensions than that of `.dcm`. Also, using the `all` boolean argument, you can specify whether the function provides output for each file, or only for the first file, which is beneficial if you are analyzing multi-slice series, as all instances have almost all the same header information. Furthermore, using the `keywords` argument you can manually specify which DICOM keywords you wish to extract. These need to be a valid keyword specified in the [DICOM standard](#).

## Usage

```
create_img_db(
    path,
    ext = c(".dcm", ".dicom", ".ima", ".tmp", ""),
    all = TRUE,
    keywords = c("StudyDate", "StudyTime", "SeriesDate", "SeriesTime", "AcquisitionDate",
                 "AcquisitionTime", "ConversionType", "Manufacturer", "InstitutionName",
                 "InstitutionalDepartmentName", "ReferringPhysicianName", "Modality",
                 "ManufacturerModelName", "StudyDescription", "SeriesDescription", "StudyComments",
                 "ProtocolName", "RequestedProcedureID", "ViewPosition", "StudyInstanceUID",
                 "SeriesInstanceUID", "SOPInstanceUID", "AccessionNumber", "PatientName", "PatientID",
                 "IssuerOfPatientID", "PatientBirthDate",
                 "PatientSex", "PatientAge",
                 "PatientSize", "PatientWeight", "StudyID", "SeriesNumber", "AcquisitionNumber",
                 "InstanceNumber", "BodyPartExamined", "SliceThickness", "SpacingBetweenSlices",
                 "PixelSpacing", "PixelAspectRatio", "Rows", "Columns", "FieldOfViewDimensions",
                 "RescaleIntercept", "RescaleSlope", "WindowCenter", "WindowWidth", "BitsAllocated",
                 "BitsStored", "PhotometricInterpretation", "KVP", "ExposureTime", "XRayTubeCurrent",
                 "ExposureInuAs", "ImageAndFluoroscopyAreaDoseProduct", "FilterType",

                 "ConvolutionKernel", "CTDIvol", "ReconstructionFieldOfView"),
    nThread = parallel::detectCores() - 1,
    na = TRUE,
    identical = TRUE
)
```

## Arguments

`path` string vector, full folder path to folder that contains the images.

ext	string array, possible file extensions to parse. It is advised to add . before the extensions as the given character patterns may be present elsewhere in the file names. Furthermore, if DICOM files without an extension should also be parsed, then add "" to the extensions as then the script will try to read all files without an extension. Also, the file names and the extensions are converted to lower case before matching to avoid mismatches due to capitals.
all	boolean, whether all files in a series should be parsed, or only the first one.
keywords	string array, of valid DICOM keywords.
nThread	integer, number of threads to use for parsing data.
na	boolean, whether to remove columns with only NA values. Defaults to <i>TRUE</i> .
identical	boolean, whether to remove columns with identical values. Defaults to <i>TRUE</i> .

### Value

data.table, with DICOM header information return unchanged. However, the function also provides additional new columns which help further data manipulations, these are:

**time\_study** POSIXct, StudyDate and StudyTime concatenated together to POSIXct.

**time\_series** POSIXct, SeriesDate and SeriesTime concatenated together to POSIXct.

**time\_acquisition** POSIXct, AcquisitionDate and AcquisitionTime concatenated together to POSIXct.

**name\_img** string, PatientName with special characters removed.

**time\_date\_of\_birth\_img** POSIXct, PatientBirthDate as POSIXct.

**img\_pixel\_spacing** numeric, PixelSpacing value of the first element in the array returned as numerical value.

### Examples

```
## Not run:
#Create a database with DICOM header information
all_dicom_headers <- create_img_db(path = "/Users/Test/Data/DICOM/")
all_dicom_headers <- create_img_db(path = "/Users/Test/Data/DICOM/", ext = c(".dcm", ".DICOM"))
#Create a database with DICOM header information for only IDs and accession numbers
all_dicom_headers <- create_img_db(path = "/Users/Test/Data/DICOM/",
keywords = c("PatientID", "AccessionNumber"))

## End(Not run)
```

---

export\_notes

*Exports free text notes to individual text files.*

---

### Description

Exports out the contents of a given cell per row into individual text files. Can be used to export out reports into individual text files for further analyses.

**Usage**

```
export_notes(d, folder, code, name1 = "ID_MERGE", name2)
```

**Arguments**

d	data.table, database containing notes loaded using the <i>load_notes</i> function. Theoretically any other data.table can be given and the contents of the specified cell will be exported into the corresponding files. In case of notes, it is advised to load them with <i>format_orig = TRUE</i> , as then the output will retain the original format of the report making it easier to read.
folder	string, full folder path to folder where the files should be exported. If folder does not exist, the function stops.
code	string vector, column name containing the data that should be exported. Generally should be <i>"abc_rep_txt"</i> , where <i>abc</i> stands for the three letter abbreviation of the given type of note.
name1	string, the first part of the file names. Defaults to <i>ID_MERGE</i> .
name2	string, the second part of the file names. <i>name1</i> and <i>name2</i> will be separated using <i>"_"</i> . Generally should be <i>"abc_rep_num"</i> , where <i>abc</i> stands for the three letter abbreviation of the given type of note.

**Value**

NULL, files are exported to given folder.

**Examples**

```
## Not run:
#Output all cardiology notes to given folder
d <- load_notes("Car.txt", type = "car", nThread = 2, format_orig = TRUE)
export_notes(d, folder = "/Users/Test/Notes/", code = "car_rep_txt",
name1 = "ID_MERGE", name2 = "car_rep_num")

## End(Not run)
```

---

find_exam	<i>Find exam data within a given timeframe using parallel CPU computing.</i>
-----------	--

---

**Description**

Finds all, earliest or closest examination to a given timepoints using parallel computing. A progress bar is also reported in the terminal to show the progress of the computation.

**Usage**

```

find_exam(
  d_from,
  d_to,
  d_from_ID = "ID_MERGE",
  d_to_ID = "ID_MERGE",
  d_from_time = "time_rad_exam",
  d_to_time = "time_enc_admit",
  time_diff_name = "timediff_exam_to_db",
  before = TRUE,
  after = TRUE,
  time = 1,
  time_unit = "days",
  multiple = "closest",
  add_column = NULL,
  keep_data = FALSE,
  nThread = parallel::detectCores() - 1,
  shared_RAM = FALSE
)

```

**Arguments**

d_from	data table, the database which is searched to find examinations within the time-frame.
d_to	data table, the database to which we wish to find examinations within the time-frame.
d_from_ID	string, column name of the patient ID column in d_from. Defaults to <i>ID_MERGE</i> .
d_to_ID	string, column name of the patient ID column in d_to. Defaults to <i>ID_MERGE</i> .
d_from_time	string, column name of the time variable column in d_from. Defaults to <i>time_rad_exam</i> .
d_to_time	string, column name of the time variable column in d_to. Defaults to <i>time_enc_admit</i> .
time_diff_name	string, column name of the new column created which holds the time difference between the exam and the time provided by d_to. Defaults to <i>timediff_exam_to_db</i> .
before	boolean, should times before the given time be considered. Defaults to <i>TRUE</i> .
after	boolean, should times after the given time be considered. Defaults to <i>TRUE</i> .
time	integer, the timeframe considered between the exam and the d_to timepoints. Defaults to <i>1</i> .
time_unit	string, the unit of time used. Time variables in d_to and d_from are truncated to the supplied time unit. For example: "2005-09-18 08:15:01 PDT" would be truncated to "2005-09-18 PDT" if <i>time_unit</i> is set to days. Then the time differences is calculated using <i>difftime</i> passing the argument to <i>units</i> . The following time units are supported: "secs", "mins", "hours", "days", "months" and "years" are supported. Defaults to <i>days</i> .
multiple	string, which exams to give back. <i>closest</i> gives back the exam closest to the time provided by d_to. <i>all</i> gives back all occurrences within the timeframe. <i>earliest</i>

	the earliest exam within the timeframe. In case of ties for <i>closest</i> or <i>earliest</i> , all are returned. Defaults to <i>closest</i> .
add_column	string, a column name in <i>d_to</i> to add to the output. Defaults to <i>NULL</i> .
keep_data	boolean, whether to include empty rows with only the <i>d_from_ID</i> column filed out for cases that have data in the <i>d_from</i> , but not within the time range. Defaults to <i>FALSE</i> .
nThread	integer, number of threads to use for parallelization. If it is set to 1, then no parallel backends are created and the function is executed sequentially.
shared_RAM	boolean, depreciated from version 1.1.0 onwards, only kept for compatibility, as Bigmemory package has issues on running on different operating systems. Now all computations are run using the memory usage specifications of the parallelization strategy.

## Value

data table, with *d\_from* filtered to ones only within the timeframe. The columns of *d\_from* are returned with the corresponding time column in *data\_to* where the rows are instances which comply with the time constraints specified by the function. An additional column specified in *time\_diff\_name* is also returned, which shows the time difference between the time column in *d\_from* and *d\_to* for that given case. Also the time column from *d\_to* specified by *d\_to\_time* is returned under the name of *time\_to\_db*. An additional column specified in *add\_column* may be added from *data\_to* to the data table.

## Examples

```
## Not run:
#Filter encounters for first emergency visits at one of MGH's ED departments
data_enc_ED <- data_enc[enc_clinic == "MGH EMERGENCY 10020010608"]
data_enc_ED <- data_enc_ED[!duplicated(data_enc_ED$ID_MERGE)]

#Find all radiological examinations within 3 day of the ED registration
rdt_ED <- find_exam(d_from = data_rdt, d_to = data_enc_ED,
  d_from_ID = "ID_MERGE", d_to_ID = "ID_MERGE",
  d_from_time = "time_rdt_exam", d_to_time = "time_enc_admit", time_diff_name = "time_diff_ED_rdt",
  before = TRUE, after = TRUE, time = 3, time_unit = "days", multiple = "all",
  nThread = 2)

#Find earliest radiological examinations within 3 day of the ED registration
rdt_ED <- find_exam(d_from = data_rdt, d_to = data_enc_ED,
  d_from_ID = "ID_MERGE", d_to_ID = "ID_MERGE",
  d_from_time = "time_rdt_exam", d_to_time = "time_enc_admit", time_diff_name = "time_diff_ED_rdt",
  before = TRUE, after = TRUE, time = 3, time_unit = "days", multiple = "earliest",
  nThread = 2)

#Find closest radiological examinations on or after 1 day of the ED registration
#and add primary diagnosis column from encounters
rdt_ED <- find_exam(d_from = data_rdt, d_to = data_enc_ED,
  d_from_ID = "ID_MERGE", d_to_ID = "ID_MERGE",
  d_from_time = "time_rdt_exam", d_to_time = "time_enc_admit", time_diff_name = "time_diff_ED_rdt",
  before = FALSE, after = TRUE, time = 1, time_unit = "days", multiple = "earliest",
```

```

add_column = "enc_diag_princ", nThread = 2)

#Find closest radiological examinations on or after 1 day of the ED registration
#but also provide empty rows for patients with exam data but not within the timeframe
rdt_ED <- find_exam(d_from = data_rdt, d_to = data_enc_ED,
d_from_ID = "ID_MERGE", d_to_ID = "ID_MERGE",
d_from_time = "time_rdt_exam", d_to_time = "time_enc_admit", time_diff_name = "time_diff_ED_rdt",
before = FALSE, after = TRUE, time = 1, time_unit = "days", multiple = "earliest",
add_column = "enc_diag_princ", keep_data = TRUE nThread = 2)

## End(Not run)

```

---

load\_all

*Loads allergy data information into R.*


---

### Description

Loads allergy information into the R environment.

### Usage

```

load_all(
  file,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
  perc = 0.6,
  na = TRUE,
  identical = TRUE,
  nThread = parallel::detectCores() - 1,
  mrn_type = FALSE
)

```

### Arguments

file	string, full file path to All.txt.
merge_id	string, column name to use to create <i>ID_MERGE</i> column used to merge different datasets. Defaults to <i>EPIC_PMRN</i> , as it is the preferred MRN in the RPDR system.
sep	string, divider between hospital ID and MRN. Defaults to <i>:</i> .
id_length	string, indicating whether to modify MRN length based-on required values <i>id_length</i> = <i>standard</i> , or to keep lengths as is <i>id_length</i> = <i>asis</i> . If <i>id_length</i> = <i>standard</i> then in case of <i>MGH</i> , <i>BWH</i> , <i>MCL</i> , <i>EMPI</i> and <i>PMRN</i> the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to <i>standard</i> .
perc	numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in <i>perc</i> x 100% of patients are kept.

na	boolean, whether to remove columns with only NA values. Defaults to <i>TRUE</i> .
identical	boolean, whether to remove columns with identical values. Defaults to <i>TRUE</i> .
nThread	integer, number of threads to use to load data.
mrn_type	boolean, should data in <i>MRN_Type</i> and <i>MRN</i> be parsed. Defaults to <i>FALSE</i> , as it is not advised to parse these for all data sources as it takes considerable time.

## Value

data table, with allergy information.

**ID\_MERGE** numeric, defined IDs by *merge\_id*, used for merging later.

**ID\_all\_EMPI** string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information from *all* datasource, corresponds to EMPI in RPDR. Data is formatted using *pretty\_mrn()*.

**ID\_all\_PMRN** string, Epic medical record number. This value is unique across Epic instances within the Partners network from *all* datasource, corresponds to EPIC\_PMRN in RPDR. Data is formatted using *pretty\_mrn()*.

**ID\_all\_loc** string, if *mrn\_type* == *TRUE*, then the data in *MRN\_Type* and *MRN* are parsed into IDs corresponding to locations (*loc*). Data is formatted using *pretty\_mrn()*.

**time\_all** POSIXct, Date when the allergy was first noted, corresponds to Noted\_Date in RPDR. Converted to POSIXct format.

**all\_all** string, Name of the allergen, corresponds to Allergen in RPDR.

**all\_all\_code** string, Epic internal identifier for the specific allergen, corresponds to Allergen\_Code in RPDR.

**all\_all\_type** string, Hierarchy for the type of allergy noted. Denotes known level of specificity of allergen, corresponds to Allergen\_Type in RPDR.

**all\_reac** string, Noted reactions to the allergen, corresponds to Reactions in RPDR.

**all\_reac\_type** string, Category of reaction to the allergen, corresponds to Reaction\_Type in RPDR.

**all\_severity** string, Degree of severity of noted reactions, corresponds to Severity in RPDR.

**all\_status** string, Last known status of allergen, either active or deleted from the patient's allergy record, corresponds to Status in RPDR.

**all\_system** string, The source system where the data was collected, corresponds to System in RPDR.

**all\_comment** string, Free-text information about the allergen, corresponds to Comments in RPDR.

**all\_del\_reason** string, Free-text information about why the allergen was removed from the patient's allergy list, corresponds to Deleted\_Reason in RPDR.

## Examples

```
## Not run:
#Using defaults
d_all <- load_all(file = "test_All.txt")

#Use sequential processing
```

```
d_all <- load_all(file = "test_All.txt", nThread = 1)

#Use parallel processing and parse data in MRN_Type and MRN columns and keep all IDs
d_all <- load_all(file = "test_All.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)
```

---

load\_all\_data

*Loads all RPDR text outputs into R.*


---

## Description

Loads all RPDR text outputs into R and returns a list of data tables processed. If multiple text files of the same type are available (if the query is larger than 25000 patients), then add a "\_" and a number to merge the same data sources into a single output in the order of the provided number.

## Usage

```
load_all_data(
  folder,
  which_data = c("mrn", "con", "dem", "all", "bib", "dia", "enc", "lab", "lno", "mcm",
    "med", "mic", "phy", "prc", "prv", "ptd", "rdt", "rfv", "trn", "car", "dis", "end",
    "hnp", "opn", "pat", "prg", "pul", "rad", "vis"),
  old_dem = FALSE,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
  perc = 0.6,
  na = TRUE,
  identical = TRUE,
  nThread = parallel::detectCores() - 1,
  many_sources = TRUE,
  load_report = TRUE,
  format_orig = FALSE
)
```

## Arguments

folder	string, full folder path to RPDR text files.
which_data	string vector, an array of abbreviation corresponding to the datasources wished to load.
old_dem	boolean, should old <i>load_dem</i> function be used for loading demographic data. Defaults to <i>TRUE</i> , should be set to <i>FALSE</i> for Dem.txt datasets prior to 2022.
merge_id	string, column name to use to create <i>ID_MERGE</i> column used to merge different datasets. Defaults to <i>EMPI</i> , as it is the preferred MRN in the RPDR system. In case of mrn dataset, leave at EMPI, as it is automatically converted to: "Enterprise_Master_Patient_Index".

sep	string, divider between hospital ID and MRN. Defaults to <code>.</code> .
id_length	string, indicating whether to modify MRN length based-on required values <code>id_length = standard</code> , or to keep lengths as is <code>id_length = asis</code> . If <code>id_length = standard</code> then in case of <i>MGH</i> , <i>BWH</i> , <i>MCL</i> , <i>EMPI</i> and <i>PMRN</i> the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to <i>standard</i> .
perc	numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in $perc \times 100\%$ of patients are kept.
na	boolean, whether to remove columns with only NA values. Defaults to <i>TRUE</i> .
identical	boolean, whether to remove columns with identical values. Defaults to <i>TRUE</i> .
nThread	integer, number of threads to use for parallelization.
many_sources	boolean, if <i>TRUE</i> , then parallelization is done on the level of the datasources. If <i>FALSE</i> , then parallelization is done within the datasources. If there are many datasources, then it is advised to set this <i>TRUE</i> , as then each different datasource will be processed in parallel. However, if there are only a few datasources selected to load, but many files per datasource (result of large queries), then it may be faster to parallelize within each datasource and therefore should be set to <i>FALSE</i> . If there are only a few sources each with one file then set to <i>TRUE</i> .
load_report	boolean, should the report text be returned for notes. Defaults to <i>TRUE</i> .
format_orig	boolean, should report be returned in its original formatting or should white spaces used for formatting be removed. Defaults to <i>FALSE</i> .

**Value**

list of parsed data tables containing the information.

**Examples**

```
## Not run:
#Load all Con, Dem and Mrn datasets processing all files within given datasource in parallel
load_all_data(folder = folder_rpdr, which_data = c("con", "dem", "mrn"),
nThread = 2, many_sources = FALSE)

#Load all supported file types parallelizing on the level of datasources
load_all_data(folder = folder_rpdr, nThread = 2, many_sources = TRUE,
format_orig = TRUE)

## End(Not run)
```

---

load\_bib

*Loads BiobankFile data into R.*


---

**Description**

Loads Biobank file data into the R environment.

**Usage**

```
load_bib(
  file,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
  perc = 0.6,
  na = TRUE,
  identical = TRUE,
  nThread = parallel::detectCores() - 1,
  mrn_type = FALSE
)
```

**Arguments**

file	string, full file path to Bib.txt.
merge_id	string, column name to use to create <i>ID_MERGE</i> column used to merge different datasets. Defaults to <i>EPIC_PMRN</i> , as it is the preferred MRN in the RPDR system.
sep	string, divider between hospital ID and MRN. Defaults to <code>:</code> .
id_length	string, indicating whether to modify MRN length based-on required values <i>id_length = standard</i> , or to keep lengths as is <i>id_length = asis</i> . If <i>id_length = standard</i> then in case of <i>MGH</i> , <i>BWH</i> , <i>MCL</i> , <i>EMPI</i> and <i>PMRN</i> the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to <i>standard</i> .
perc	numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in <i>perc x 100%</i> of patients are kept. Not used for loading mrn data.
na	boolean, whether to remove columns with only NA values. Defaults to <i>TRUE</i> .
identical	boolean, whether to remove columns with identical values. Defaults to <i>TRUE</i> .
nThread	integer, number of threads to use to load data.
mrn_type	boolean, should data in <i>MRN_Type</i> and <i>MRN</i> be parsed. Defaults to <i>FALSE</i> , as it is not advised to parse these for all data sources as it takes considerable time.

**Value**

data table, with BiobankFile data.

**ID\_MERGE** numeric, defined IDs by *merge\_id*, used for merging later.

**ID\_bib\_PMRN** string, Epic medical record number. This value is unique across Epic instances within the Partners network, corresponds to *EPIC\_PMRN* in RPDR. Data is formatted using `pretty_mrn()`.

**ID\_bib\_EMPI** string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information, corresponds to *Enterprise\_Master\_Patient\_Index* in RPDR. Data is formatted using `pretty_mrn()`.

**ID\_bib\_MGH** string, Unique Medical Record Number for Mass General Hospital, corresponds to *MGH\_MRN* in RPDR. Data is formatted using `pretty_mrn()`.

- ID\_bib\_BWH** string, Unique Medical Record Number for Brigham and Women's Hospital, corresponds to BWH\_MRN in RPDR. Data is formatted using pretty\_mrn().
- ID\_bib\_FH** string, Unique Medical Record Number for Faulkner Hospital, corresponds to FH\_MRN in RPDR. Data is formatted using pretty\_mrn().
- ID\_bib\_SRH** string, Unique Medical Record Number for Spaulding Rehabilitation Hospital, corresponds to SRH\_MRN in RPDR. Data is formatted using pretty\_mrn().
- ID\_bib\_NWH** string, Unique Medical Record Number for Newton-Wellesley Hospital, corresponds to NWH\_MRN in RPDR. Data is formatted using pretty\_mrn().
- ID\_bib\_NSMC** string, Unique Medical Record Number for North Shore Medical Center, corresponds to NSMC\_MRN in RPDR. Data is formatted using pretty\_mrn().
- ID\_bib\_MCL** string, Unique Medical Record Number for McLean Hospital, corresponds to MCL\_MRN in RPDR. Data is formatted using pretty\_mrn().
- ID\_bib\_MEE** string, Unique Medical Record Number for Mass Eye and Ear, corresponds to MEE\_MRN in RPDR. Data is formatted using pretty\_mrn().
- ID\_bib\_DFC** string, Unique Medical Record Number for Dana Farber Cancer center, corresponds to DFC\_MRN in RPDR. Data is formatted using pretty\_mrn(). Legacy data.
- ID\_bib\_WDH** string, Unique Medical Record Number for Wentworth-Douglass Hospital, corresponds to WDH\_MRN in RPDR. Data is formatted using pretty\_mrn(). Legacy data.
- bib\_subject\_ID** string, Biobank unique patient identifier, corresponds to Subject\_ID in RPDR. ID is not formatted.
- bib\_subject\_ID** string, This will always default to Biobank, corresponds to Registry Name in RPDR.

### Examples

```
## Not run:
#Using defaults
d_bib <- load_bib(file = "test_Bib.txt")

#Use sequential processing
d_bib <- load_bib(file = "test_Bib.txt", nThread = 1)

#Use parallel processing and parse data in MRN_Type and MRN columns and keep all IDs
d_bib <- load_bib(file = "test_Bib.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)
```

---

load\_con

*Loads contact information into R.*

---

### Description

Loads patient contact, insurance, and PCP information into the R environment.

**Usage**

```
load_con(
  file,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
  perc = 0.6,
  na = TRUE,
  identical = TRUE,
  nThread = parallel::detectCores() - 1,
  mrn_type = TRUE
)
```

**Arguments**

file	string, full file path to Con.txt.
merge_id	string, column name to use to create <i>ID_MERGE</i> column used to merge different datasets. Defaults to <i>EPIC_PMRN</i> , as it is the preferred MRN in the RPDR system.
sep	string, divider between hospital ID and MRN. Defaults to <code>:</code> .
id_length	string, indicating whether to modify MRN length based-on required values <i>id_length</i> = <i>standard</i> , or to keep lengths as is <i>id_length</i> = <i>asis</i> . If <i>id_length</i> = <i>standard</i> then in case of <i>MGH</i> , <i>BWH</i> , <i>MCL</i> , <i>EMPI</i> and <i>PMRN</i> the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to <i>standard</i> .
perc	numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in <i>perc</i> x 100% of patients are kept.
na	boolean, whether to remove columns with only NA values. Defaults to <i>TRUE</i> .
identical	boolean, whether to remove columns with identical values. Defaults to <i>TRUE</i> .
nThread	integer, number of threads to use to load data.
mrn_type	boolean, should data in <i>MRN_Type</i> and <i>MRN</i> be parsed. Defaults to <i>TRUE</i> only for Con.txt, as it is not advised to parse these for all data sources as it takes considerable time.

**Value**

data table, with contact information data.

**ID\_MERGE** numeric, defined IDs by *merge\_id*, used for merging later.

**ID\_con\_EMPI** string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information from *con* datasource, corresponds to EMPI in RPDR. Data is formatted using `pretty_mrn()`.

**ID\_con\_PMRN** string, Epic medical record number. This value is unique across Epic instances within the Partners network from *con*datasource, corresponds to EPIC\_PMRN in RPDR. Data is formatted using `pretty_mrn()`.

- ID\_con\_loc** string, if `mrn_type == TRUE`, then the data in *MRN\_Type* and *MRN* are parsed into IDs corresponding to locations (*loc*). Data is formatted using `pretty_mrn()`.
- ID\_con\_loc\_list** string, if prevalence of IDs in *Patient\_ID\_List* > *perc*, then they are included in the output. Data is formatted using `pretty_mrn()`.
- name\_last** string, Patient's last name, corresponds to *Last\_Name* in RPDR.
- name\_first** string, Patient's first name, corresponds to *First\_Name* in RPDR.
- name\_middle** string, Patient's middle name or initial, corresponds to *Middle\_Name* in RPDR.
- name\_previous** string, Any alternate names on record for this patient, corresponds to *Previous\_Name* in RPDR.
- SSN** string, Social Security Number, corresponds to *SSN* in RPDR.
- VIP** character, Special patient statuses as defined by the EMPI group, corresponds to *VIP* in RPDR.
- address1** string, Patient's current address, corresponds to *address1* in RPDR.
- address2** string, Additional address information, corresponds to *address2* in RPDR.
- city** string, City of residence, corresponds to *City* in RPDR.
- state** string, State of residence, corresponds to *State* in RPDR.
- country\_con** string, Country of residence from con datasource, corresponds to *Country* in RPDR.
- zip\_con** numeric, Mailing zip code of primary residence from con datasource, corresponds to *Zip* in RPDR. Formatted to 5 character zip codes using `pretty_numbers()`.
- direct\_contact\_consent** boolean, Indicates whether the patient has given permission to contact them directly through the RODY program, corresponds to *Direct\_Contact\_Consent* in RPDR. Legacy variable.
- research\_invitations** boolean, Indicates if a patient can be invited to participate in research, corresponds to *Research\_Invitations* in RPDR.
- phone\_home** number, Patient's home phone number, corresponds to *Home\_Phone* in RPDR. Formatted to 10 digit phone numbers using `pretty_numbers()`.
- phone\_day** number, Phone number where the patient can be reached during the day, corresponds to *Day\_Phone* in RPDR. Formatted to 10 digit phone numbers using `pretty_numbers()`.
- insurance1** string, Patient's primary health insurance carrier and subscriber ID information, corresponds to *Insurance\_1* in RPDR.
- insurance2** string, Patient's secondary health insurance carrier and subscriber ID information, if any, corresponds to *Insurance\_2* in RPDR.
- insurance3** string, Patient's tertiary health insurance carrier and subscriber ID information, if any, corresponds to *Insurance\_3* in RPDR.
- primary\_care\_physician** string, Comma-delimited list of all primary care providers on record for this patient per institution, along with contact information (if available), corresponds to *Primary\_Care\_Physician* in RPDR.
- primary\_care\_physician\_resident** string, Comma-delimited list of any Resident primary care providers on record for this patient per institution, along with contact information (if available), corresponds to *Resident\_Primary\_Care\_Physician* in RPDR.

## Examples

```
## Not run:
#Using defaults
d_con <- load_con(file = "test_Con.txt")

#Use sequential processing
d_con <- load_con(file = "test_Con.txt", nThread = 1)

#Use parallel processing and parse data in
#MRN_Type and MRN columns (default in load_con) and keep all IDs
d_con <- load_con(file = "test_Con.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)
```

---

load_dem	<i>Loads demographic information into R for new demographic tables following changes in the beginning of 2022.</i>
----------	--

---

## Description

Loads patient demographic and vital status information into the R environment. Since version 0.2.2 of the software this function supports the new demographics table data definitions.

## Usage

```
load_dem(
  file,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
  perc = 0.6,
  na = TRUE,
  identical = TRUE,
  nThread = parallel::detectCores() - 1,
  mrn_type = FALSE
)
```

## Arguments

file	string, full file path to Dem.txt.
merge_id	string, column name to use to create <i>ID_MERGE</i> column used to merge different datasets. Defaults to <i>EPIC_PMRN</i> , as it is the preferred MRN in the RPDR system.
sep	string, divider between hospital ID and MRN. Defaults to <code>:</code> .

id_length	string, indicating whether to modify MRN length based-on required values <i>id_length</i> = <i>standard</i> , or to keep lengths as is <i>id_length</i> = <i>asis</i> . If <i>id_length</i> = <i>standard</i> then in case of <i>MGH</i> , <i>BWH</i> , <i>MCL</i> , <i>EMPI</i> and <i>PMRN</i> the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to <i>standard</i> .
perc	numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in <i>perc</i> x 100% of patients are kept.
na	boolean, whether to remove columns with only NA values. Defaults to <i>TRUE</i> .
identical	boolean, whether to remove columns with identical values. Defaults to <i>TRUE</i> .
nThread	integer, number of threads to use to load data.
mrn_type	boolean, should data in <i>MRN_Type</i> and <i>MRN</i> be parsed. Defaults to <i>FALSE</i> , as it is not advised to parse these for all data sources as it takes considerable time.

### Value

data table, with demographic information data.

**ID\_MERGE** numeric, defined IDs by *merge\_id*, used for merging later.

**ID\_dem\_EMPI** string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information. from *dem* datasource, corresponds to EMPI in RPDR. Data is formatted using *pretty\_mrn()*.

**ID\_dem\_PMRN** string, Epic medical record number. This value is unique across Epic instances within the Partners network. from *dem* datasource, corresponds to EPIC\_PMRN in RPDR. Data is formatted using *pretty\_mrn()*.

**ID\_dem\_loc** string, if *mrn\_type* == *TRUE*, then the data in *MRN\_Type* and *MRN* are parsed into IDs corresponding to locations (*loc*). Data is formatted using *pretty\_mrn()*.

**gender\_legal\_sex** string, Patient's legal sex, corresponds to Gender\_Legal\_Sex in RPDR.

**sex\_at\_birth** string, Patient's sex at time of birth, corresponds to Sex\_at\_Birth in RPDR.

**gender\_identity** string, Patient's personal conception of their gender, corresponds to Gender\_Identity in RPDR.

**time\_date\_of\_birth** POSIXct, Patient's date of birth, corresponds to Date\_of\_Birth. Converted to POSIXct format.

**age** string, Patient's current age (or age at death), corresponds to Age in RPDR.

**language** string, Patient's preferred spoken language, corresponds to Language in RPDR.

**language\_group** string, Patient's preferred language: English or Non-English, corresponds to Language\_Group in RPDR.

**race\_1** string, Patient's primary race, corresponds to Race1 in RPDR.

**race\_2** string, Patient's primary race if more than one race, corresponds to Race2 in RPDR.

**race\_group** string, Patient's Race Group as determined by Race1 and Race2, corresponds to Race\_Group in RPDR.

**ethnic\_group** string, Patient's Ethnicity: Hispanic or Non Hispanic, corresponds to Ethnic\_Group in RPDR.

**marital** string, Patient's current marital status, corresponds to Marital\_Status in RPDR.

**religion** string, Patient-identified religious preference, corresponds to Religion in RPDR.

**veteran** string, Patient's current military veteran status, corresponds to Is\_a\_veteran in RPDR.

**country\_dem** string, Patient's current country of residence from dem datasource, corresponds to Country in RPDR.

**zip\_dem** string, Mailing zip code of patient's primary residence from dem datasource, corresponds to Zip\_code in RPDR. Formatted to 5 character zip codes.

**vital\_status** string, Identifies if the patient is living or deceased. This data is updated monthly from the Partners registration system and the Social Security Death Master Index, corresponds to Vital\_Status in RPDR. Punctuation marks are removed.

**time\_date\_of\_death** POSIXct, Recorded date of death from source in 'Vital\_Status'. Date of death information obtained solely from the Social Security Death Index will not be reported until 3 years after death due to privacy concerns. If the value is independently documented by a Partners entity within the 3 year window then the date will be displayed. corresponds to Date\_of\_Death in RPDR. Converted to POSIXct format.

### Examples

```
## Not run:
#Using defaults
d_dem <- load_dem(file = "test_Dem.txt")

#Use sequential processing
d_dem <- load_dem(file = "test_Dem.txt", nThread = 1)

#Use parallel processing and parse data in MRN_Type and MRN columns and keep all IDs
d_dem <- load_dem(file = "test_Dem.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)
```

---

load_dem_old	<i>Loads demographic information into R for demographics tables before 2022.</i>
--------------	--

---

### Description

Loads patient demographic and vital status information into the R environment. Since version 0.2.2 of the software, this function supports the old demographics table data definitions and is identical to the *load\_dem* function of previous versions of the software.

### Usage

```
load_dem_old(
  file,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
  perc = 0.6,
```

```

na = TRUE,
identical = TRUE,
nThread = parallel::detectCores() - 1,
mrn_type = FALSE
)

```

### Arguments

file	string, full file path to Dem.txt.
merge_id	string, column name to use to create <i>ID_MERGE</i> column used to merge different datasets. Defaults to <i>EPIC_PMRN</i> , as it is the preferred MRN in the RPDR system.
sep	string, divider between hospital ID and MRN. Defaults to <code>.</code> .
id_length	string, indicating whether to modify MRN length based-on required values <i>id_length</i> = <i>standard</i> , or to keep lengths as is <i>id_length</i> = <i>asis</i> . If <i>id_length</i> = <i>standard</i> then in case of <i>MGH</i> , <i>BWH</i> , <i>MCL</i> , <i>EMPI</i> and <i>PMRN</i> the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to <i>standard</i> .
perc	numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in <i>perc</i> x 100% of patients are kept.
na	boolean, whether to remove columns with only NA values. Defaults to <i>TRUE</i> .
identical	boolean, whether to remove columns with identical values. Defaults to <i>TRUE</i> .
nThread	integer, number of threads to use to load data.
mrn_type	boolean, should data in <i>MRN_Type</i> and <i>MRN</i> be parsed. Defaults to <i>FALSE</i> , as it is not advised to parse these for all data sources as it takes considerable time.

### Value

data table, with demographic information data.

**ID\_MERGE** numeric, defined IDs by *merge\_id*, used for merging later.

**ID\_dem\_EMPI** string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information. from *dem* datasource, corresponds to EMPI in RPDR. Data is formatted using `pretty_mrn()`.

**ID\_dem\_PMRN** string, Epic medical record number. This value is unique across Epic instances within the Partners network. from *dem* datasource, corresponds to EPIC\_PMRN in RPDR. Data is formatted using `pretty_mrn()`.

**ID\_dem\_loc** string, if *mrn\_type* == *TRUE*, then the data in *MRN\_Type* and *MRN* are parsed into IDs corresponding to locations (*loc*). Data is formatted using `pretty_mrn()`.

**gender** string, Patient's legal sex, corresponds to Gender in RPDR.

**time\_date\_of\_birth** POSIXct, Patient's date of birth, corresponds to Date\_of\_Birth in RPDR. Converted to POSIXct format.

**age** string, Patient's current age (or age at death), corresponds to Age in RPDR.

**language** string, Patient's preferred spoken language, corresponds to Language in RPDR.

**race** string, Patient's primary race, corresponds to Race in RPDR.

**marital** string, Patient's current marital status, corresponds to Marital\_Status in RPDR.

**religion** string, Patient-identified religious preference, corresponds to Religion in RPDR.

**veteran** string, Patient's current military veteran status, corresponds to Is\_a\_veteran in RPDR.

**country\_dem** string, Patient's current country of residence from dem datasource, corresponds to Country in RPDR.

**zip\_dem** string, Mailing zip code of patient's primary residence from dem datasource, corresponds to Zip\_code in RPDR. Formatted to 5 character zip codes.

**vital\_status** string, Identifies if the patient is living or deceased. This data is updated monthly from the Partners registration system and the Social Security Death Master Index, corresponds to Vital\_Status in RPDR. Punctuation marks are removed.

**time\_date\_of\_death** POSIXct, Recorded date of death from source in 'Vital\_Status'. Date of death information obtained solely from the Social Security Death Index will not be reported until 3 years after death due to privacy concerns. If the value is independently documented by a Partners entity within the 3 year window then the date will be displayed. corresponds to Date\_of\_Death in RPDR. Converted to POSIXct format.

### Examples

```
## Not run:
#Using defaults
d_dem <- load_dem_old(file = "test_Dem.txt")

#Use sequential processing
d_dem <- load_dem_old(file = "test_Dem.txt", nThread = 1)

#Use parallel processing and parse data in MRN_Type and MRN columns and keep all IDs
d_dem <- load_dem_old(file = "test_Dem.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)
```

---

load\_dia

*Loads diagnoses into R.*

---

### Description

Loads diagnoses information into the R environment, both Dia and Dea files.

### Usage

```
load_dia(
  file,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
  perc = 0.6,
```

```

na = TRUE,
identical = TRUE,
nThread = parallel::detectCores() - 1,
mrn_type = FALSE
)

```

### Arguments

file	string, full file path to Dia.txt or Dea.txt.
merge_id	string, column name to use to create <i>ID_MERGE</i> column used to merge different datasets. Defaults to <i>EPIC_PMRN</i> , as it is the preferred MRN in the RPDR system.
sep	string, divider between hospital ID and MRN. Defaults to <code>.</code> .
id_length	string, indicating whether to modify MRN length based-on required values <i>id_length = standard</i> , or to keep lengths as is <i>id_length = asis</i> . If <i>id_length = standard</i> then in case of <i>MGH</i> , <i>BWH</i> , <i>MCL</i> , <i>EMPI</i> and <i>PMRN</i> the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to <i>standard</i> .
perc	numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in <i>perc x 100%</i> of patients are kept.
na	boolean, whether to remove columns with only NA values. Defaults to <i>TRUE</i> .
identical	boolean, whether to remove columns with identical values. Defaults to <i>TRUE</i> .
nThread	integer, number of threads to use to load data.
mrn_type	boolean, should data in <i>MRN_Type</i> and <i>MRN</i> be parsed. Defaults to <i>FALSE</i> , as it is not advised to parse these for all data sources as it takes considerable time.

### Value

data table, with diagnoses information.

**ID\_MERGE** numeric, defined IDs by *merge\_id*, used for merging later.

**ID\_dia\_EMPI** string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information from *dia* datasource, corresponds to EMPI in RPDR. Data is formatted using *pretty\_mrn()*.

**ID\_dia\_PMRN** string, Epic medical record number. This value is unique across Epic instances within the Partners network from *dia* datasource, corresponds to EPIC\_PMRN in RPDR. Data is formatted using *pretty\_mrn()*.

**ID\_dia\_loc** string, if *mrn\_type == TRUE*, then the data in *MRN\_Type* and *MRN* are parsed into IDs corresponding to locations (*loc*). Data is formatted using *pretty\_mrn()*.

**time\_dia** POSIXct, Date when the diagnosis was noted, corresponds to Date in RPDR. Converted to POSIXct format.

**dia\_name** string, Name of the diagnosis, diagnosis-related group, or phenotype. For more information on available Phenotypes visit [https://phenotypes.partners.org/phenotype\\_list.html](https://phenotypes.partners.org/phenotype_list.html), corresponds to *Diagnosis\_Name* in RPDR.

**dia\_code** string, Diagnosis, diagnosis-related group, or phenotype code, corresponds to Code in RPDR.

**dia\_code\_type** string, Standardized classification system or custom grouping associated with the diagnosis code, corresponds to Code\_type in RPDR.

**dia\_flag** string, Qualifier for the diagnosis, if any, corresponds to Diagnosis\_flag in RPDR.

**dia\_enc\_num** string, Unique identifier of the record/visit. This values includes the source system, hospital, and a unique identifier within the source system, corresponds to Encounter\_number in RPDR.

**dia\_provider** string, Provider of record for the encounter where the diagnosis was entered, corresponds to Provider in RPDR.

**dia\_clinic** string, Specific department/location where the patient encounter took place, corresponds to Clinic in RPDR.

**dia\_hosp** string, Facility where the encounter occurred, corresponds to Hospital in RPDR.

**dia\_inpatient** string, Identifies whether the diagnosis was noted during an inpatient or outpatient encounter, corresponds to Inpatient\_Outpatient in RPDR. Punctuation marks removed.

### Examples

```
## Not run:
#Using defaults
d_dia <- load_dia(file = "test_Dia.txt")

#Use sequential processing
d_dia <- load_dia(file = "test_Dia.txt", nThread = 1)

#Use parallel processing and parse data in MRN_Type and MRN columns and keep all IDs
d_dea <- load_dia(file = "test_Dea.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)
```

---

load\_enc

*Loads encounter information into R.*

---

### Description

Loads encounter-level detail information into the R environment, both Enc and Exc files.

### Usage

```
load_enc(
  file,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
  perc = 0.6,
  na = TRUE,
```

```

    identical = TRUE,
    nThread = parallel::detectCores() - 1,
    mrn_type = FALSE
)

```

### Arguments

file	string, full file path to Enc.txt or Exc.txt
merge_id	string, column name to use to create <i>ID_MERGE</i> column used to merge different datasets. Defaults to <i>EPIC_PMRN</i> , as it is the preferred MRN in the RPDR system.
sep	string, divider between hospital ID and MRN. Defaults to <code>.</code>
id_length	string, indicating whether to modify MRN length based-on required values <i>id_length = standard</i> , or to keep lengths as is <i>id_length = asis</i> . If <i>id_length = standard</i> then in case of <i>MGH</i> , <i>BWH</i> , <i>MCL</i> , <i>EMPI</i> and <i>PMRN</i> the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to <i>standard</i> .
perc	numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in <i>perc x 100%</i> of patients are kept.
na	boolean, whether to remove columns with only NA values. Defaults to <i>TRUE</i> .
identical	boolean, whether to remove columns with identical values. Defaults to <i>TRUE</i> .
nThread	integer, number of threads to use to load data.
mrn_type	boolean, should data in <i>MRN_Type</i> and <i>MRN</i> be parsed. Defaults to <i>FALSE</i> , as it is not advised to parse these for all data sources as it takes considerable time.

### Value

data table, with encounter information.

**ID\_MERGE** numeric, defined IDs by *merge\_id*, used for merging later.

**ID\_enc\_EMPI** string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information from *enc* datasource, corresponds to EMPI in RPDR. Data is formatted using *pretty\_mrn()*.

**ID\_enc\_PMRN** string, Epic medical record number. This value is unique across Epic instances within the Partners network from *enc* datasource, corresponds to EPIC\_PMRN in RPDR. Data is formatted using *pretty\_mrn()*.

**ID\_enc\_loc** string, if *mrn\_type == TRUE*, then the data in *MRN\_Type* and *MRN* are parsed into IDs corresponding to locations (*loc*). Data is formatted using *pretty\_mrn()*.

**enc\_num** string, Unique identifier of the record/visit. This values includes the source system, hospital, and a unique identifier within the source system, corresponds to *Encounter\_number* in RPDR.

**time\_enc\_admit** POSIXct, Date when the patient was admitted or entered the facility, corresponds to *Admit\_Date* in RPDR. Converted to POSIXct format.

**time\_enc\_disch** POSIXct, Date when the patient was discharged or left the facility, corresponds to *Discharge\_Date* in RPDR. Converted to POSIXct format.

- enc\_status** string, Billing account-related notes about the encounter. This will not be populated for all encounters, corresponds to Encounter\_Status in RPDR.
- enc\_hosp** string, Facility where the encounter occurred, corresponds to Hospital in RPDR.
- enc\_inpatient** string, Classifies the type of encounter as either Inpatient or Outpatient. ED visits are currently classified under the 'Outpatient' label, corresponds to Inpatient\_or\_Outpatient in RPDR.
- enc\_service** string, Hospital service line assigned to the encounter, corresponds to Service\_Line in RPDR.
- enc\_attending** string, The attending provider associated with the encounter. For Epic professional billing, this is the billing provider, corresponds to Attending\_MD in RPDR.
- enc\_length** numeric, Length of stay for the encounter, corresponds to LOS\_days in RPDR.
- enc\_clinic** string, Specific department/location where the encounter occurred, corresponds to Clinic\_Name in RPDR.
- enc\_admit\_src** string, Location where the patient was admitted when entering the hospital/clinic, corresponds to Admit\_Source in RPDR.
- enc\_pat\_type** string, Provides information regarding the specific patient classifications and status of the patient visit. This field is only populated for McLean Hospital encounters, corresponds to Patient\_Type in RPDR.
- enc\_ref\_disp** string, Location where the patient has been directed for treatment or follow-up by a staff member. This field is only populated for McLean Hospital encounters, corresponds to Referrer\_Discipline in RPDR.
- enc\_disch\_disp** string, Patient's anticipated location or status following the encounter, corresponds to Discharge\_Disposition in RPDR.
- enc\_pay** string, Payors responsible for the hospital account. Multiple payors (primary, secondary, etc.) may be listed, corresponds to Payor in RPDR.
- enc\_diag\_admit** string, Initial working diagnosis documented by the admitting or attending physician, corresponds to Admitting\_Diagnosis in RPDR.
- enc\_diag\_princ** string, Condition established, after study, to be chiefly responsible for occasioning the admission of the patient to the hospital for care, corresponds to Principle\_Diagnosis in RPDR.
- enc\_diag\_1** string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis\_1 in RPDR.
- enc\_diag\_2** string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis\_2 in RPDR.
- enc\_diag\_3** string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis\_3 in RPDR.
- enc\_diag\_4** string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis\_4 in RPDR.
- enc\_diag\_5** string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis\_5 in RPDR.
- enc\_diag\_6** string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis\_6 in RPDR.

**enc\_diag\_7** string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis\_7 in RPDR.

**enc\_diag\_8** string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis\_8 in RPDR.

**enc\_diag\_9** string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis\_9 in RPDR.

**enc\_diag\_10** string, Additional diagnoses associated with this encounter or visit, corresponds to Diagnosis\_10 in RPDR.

**enc\_diag\_group** string, Diagnosis-Related Group for the encounter, in the following format: SYSTEM:CODE - Description, corresponds to DRG in RPDR.

### Examples

```
## Not run:
#Using defaults
d_enc <- load_enc(file = "test_Enc.txt")

#Use sequential processing
d_enc <- load_enc(file = "test_Enc.txt", nThread = 1)

#Use parallel processing and parse data in MRN_Type and MRN columns and keep all IDs
d_exc <- load_enc(file = "test_Exc.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)
```

---

load\_lab

*Loads laboratory results into R.*

---

### Description

Loads laboratory results into the R environment, both Lab and Clb files.

### Usage

```
load_lab(
  file,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
  perc = 0.6,
  na = TRUE,
  identical = TRUE,
  nThread = parallel::detectCores() - 1,
  mrn_type = FALSE
)
```

**Arguments**

file	string, full file path to Lab.txt or Clb.txt.
merge_id	string, column name to use to create <i>ID_MERGE</i> column used to merge different datasets. Defaults to <i>EPIC_PMRN</i> , as it is the preferred MRN in the RPDR system.
sep	string, divider between hospital ID and MRN. Defaults to <code>:</code> .
id_length	string, indicating whether to modify MRN length based-on required values <i>id_length</i> = <i>standard</i> , or to keep lengths as is <i>id_length</i> = <i>asis</i> . If <i>id_length</i> = <i>standard</i> then in case of <i>MGH</i> , <i>BWH</i> , <i>MCL</i> , <i>EMPI</i> and <i>PMRN</i> the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to <i>standard</i> .
perc	numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in <i>perc</i> x 100% of patients are kept.
na	boolean, whether to remove columns with only NA values. Defaults to <i>TRUE</i> .
identical	boolean, whether to remove columns with identical values. Defaults to <i>TRUE</i> .
nThread	integer, number of threads to use to load data.
mrn_type	boolean, should data in <i>MRN_Type</i> and <i>MRN</i> be parsed. Defaults to <i>FALSE</i> , as it is not advised to parse these for all data sources as it takes considerable time.

**Value**

data table, with laboratory exam information.

**ID\_MERGE** numeric, defined IDs by *merge\_id*, used for merging later.

**ID\_lab\_EMPI** string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information from *lab* datasource, corresponds to EMPI in RPDR. Data is formatted using `pretty_mrn()`.

**ID\_lab\_PMRN** string, Epic medical record number. This value is unique across Epic instances within the Partners network from *lab* datasource, corresponds to EPIC\_PMRN in RPDR. Data is formatted using `pretty_mrn()`.

**ID\_lab\_loc** string, if *mrn\_type* == *TRUE*, then the data in *MRN\_Type* and *MRN* are parsed into IDs corresponding to locations (*loc*). Data is formatted using `pretty_mrn()`.

**time\_lab\_result** POSIXct, Date when the specimen was collected, corresponds to Seq\_Date\_Time in RPDR. Converted to POSIXct format.

**lab\_group** string, Higher-level grouping concept used to consolidate similar tests across hospitals, corresponds to Group\_ID in RPDR.

**lab\_loinc** string, Standardized LOINC code for the laboratory test, corresponds to Loinc\_Code in RPDR.

**lab\_testID** string, Internal identifier for the test used by the source system, corresponds to Test\_ID in RPDR.

**lab\_descript** string, Name of the lab test, corresponds to Test\_Description in RPDR.

**lab\_result** string, Result value for the test, corresponds to Result in RPDR.

- lab\_result\_txt** string, Additional information included with the result. This can include instructions for interpretation or comments from the laboratory, corresponds to Result\_Text in RPDR.
- lab\_result\_abn** string, Flag for identifying if values are outside of normal ranges or represent a significant deviation from previous values, corresponds to Abnormal\_Flag in RPDR.
- lab\_result\_unit** string, Units associated with the result value, corresponds to Reference\_Unit in RPDR.
- lab\_result\_range** string, Normal or therapeutic range for this value, corresponds to Reference\_Range in RPDR.
- lab\_result\_toxic** string, Reference range of values defined as being toxic to the patient, corresponds to Toxic\_Range in RPDR.
- lab\_spec** string, Type of specimen collected to perform the test, corresponds to Specimen\_Type in RPDR.
- lab\_spec\_txt** string, Free-text information about the specimen, its collection or its integrity, corresponds to Specimen\_Text in RPDR.
- lab\_correction** string, Free-text information about any changes made to the results, corresponds to Correction\_Flag in RPDR.
- lab\_status** string, Flag which indicates whether the procedure is pending or complete, corresponds to Test\_Status in RPDR.
- lab\_ord\_pys** string, Name of the ordering physician, corresponds to Ordering\_Doc in RPDR.
- lab\_accession** string, Internal tracking number assigned to the specimen for identification in the lab, corresponds to Accession in RPDR.
- lab\_source** string, Database source, either CDR (Clinical Data Repository) or RPDR (internal RPDR database), corresponds to Source in RPDR.

### Examples

```
## Not run:
#Using defaults
d_lab <- load_lab(file = "test_Lab.txt")

#Use sequential processing
d_lab <- load_lab(file = "test_Lab.txt", nThread = 1)

#Use parallel processing and parse data in MRN_Type and MRN columns and keep all IDs
d_clb <- load_lab(file = "test_Clb.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)
```

---

load\_lno

*Loads LMR note documents into R.*

---

### Description

Loads notes from the LMR legacy EHR system.

**Usage**

```
load_lno(
  file,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
  perc = 0.6,
  na = TRUE,
  identical = TRUE,
  nThread = parallel::detectCores() - 1,
  mrn_type = FALSE
)
```

**Arguments**

file	string, full file path to Lno.txt.
merge_id	string, column name to use to create <i>ID_MERGE</i> column used to merge different datasets. Defaults to <i>EPIC_PMRN</i> , as it is the preferred MRN in the RPDR system.
sep	string, divider between hospital ID and MRN. Defaults to <code>:</code> .
id_length	string, indicating whether to modify MRN length based-on required values <i>id_length = standard</i> , or to keep lengths as is <i>id_length = asis</i> . If <i>id_length = standard</i> then in case of <i>MGH</i> , <i>BWH</i> , <i>MCL</i> , <i>EMPI</i> and <i>PMRN</i> the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to <i>standard</i> .
perc	numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in <i>perc x 100%</i> of patients are kept.
na	boolean, whether to remove columns with only NA values. Defaults to <i>TRUE</i> .
identical	boolean, whether to remove columns with identical values. Defaults to <i>TRUE</i> .
nThread	integer, number of threads to use to load data.
mrn_type	boolean, should data in <i>MRN_Type</i> and <i>MRN</i> be parsed. Defaults to <i>FALSE</i> , as it is not advised to parse these for all data sources as it takes considerable time.

**Value**

data table, with LMR notes information.

**ID\_MERGE** numeric, defined IDs by *merge\_id*, used for merging later.

**ID\_lno\_EMPI** string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information from *lno* datasource, corresponds to EMPI in RPDR. Data is formatted using *pretty\_mrn()*.

**ID\_lno\_PMRN** string, Epic medical record number. This value is unique across Epic instances within the Partners network from *lno* datasource, corresponds to EPIC\_PMRN in RPDR. Data is formatted using *pretty\_mrn()*.

**ID\_lno\_loc** string, if *mrn\_type == TRUE*, then the data in *MRN\_Type* and *MRN* are parsed into IDs corresponding to locations (*loc*). Data is formatted using *pretty\_mrn()*.

**time\_lno** POSIXct, Date when the report was filed, corresponds to LMRNote\_Date in RPDR. Converted to POSIXct format.

**lno\_rec\_id** string, Internal identifier for this report within the LMR system, corresponds to Record\_Id in RPDR.

**lno\_status** string, Completion status of the note, corresponds to Status in RPDR.

**lno\_author** string, Name of user who created the note, corresponds to Author in RPDR.

**lno\_author\_mrn** string, Author's user identifier within the LMR system, corresponds to Author\_MRN in RPDR.

**lno\_COD** string, Hospital-specific user code of the note author. The first character is a hospital-specific prefix, corresponds to COD in RPDR.

**lno\_hosp** string, Facility where the encounter occurred, corresponds to Institution in RPDR.

**lno\_subject** string, Type of note. This value is derived from the "Subject" line of the narrative text, corresponds to Subject in RPDR.

**lno\_rep\_txt** string, Full narrative text of the note, corresponds to Comments in RPDR.

### Examples

```
## Not run:
#Using defaults
d_lno <- load_lno(file = "test_Lno.txt")

#Use sequential processing
d_lno <- load_lno(file = "test_Lno.txt", nThread = 1)

#Use parallel processing and parse data in MRN_Type and MRN columns and keep all IDs
d_lno <- load_lno(file = "test_Lno.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)
```

---

load\_mcm

*Loads match control data into R.*

---

### Description

Loads match control tables into the R environment.

### Usage

```
load_mcm(
  file,
  sep = ":",
  id_length = "standard",
  na = TRUE,
  identical = TRUE,
  nThread = parallel::detectCores() - 1
)
```

**Arguments**

file	string, full file path to Mcm.txt.
sep	string, divider between hospital ID and MRN. Defaults to <code>.</code> .
id_length	string, indicating whether to modify MRN length based-on required values <i>id_length</i> = <i>standard</i> , or to keep lengths as is <i>id_length</i> = <i>asis</i> .
na	boolean, whether to remove columns with only NA values. Defaults to <i>TRUE</i> .
identical	boolean, whether to remove columns with identical values. Defaults to <i>TRUE</i> .
nThread	integer, number of threads to use to load data.

**Value**

data table, with matching data.

**ID\_case\_PMRN** string, Epic PMRN value for a patient in the index cohort, corresponds to Case\_Patient\_EPIC\_PMRN in RPDR. Data is formatted using `pretty_mrn()`.

**ID\_case\_EMPI** string, EMPI value for a patient in the index cohort, corresponds to Case\_Patient\_EMPI in RPDR. Data is formatted using `pretty_mrn()`.

**ID\_control\_PMRN** string, Epic PMRN value for a patient matched to a case in the index cohort, corresponds to Control\_Patient\_EPIC\_PMRN in RPDR. Data is formatted using `pretty_mrn()`.

**ID\_control\_EMPI** string, EMPI value for a control patient matched to a case in the index cohort, corresponds to Control\_Patient\_EMPI in RPDR. Data is formatted using `pretty_mrn()`.

**match\_strength** string, Number of similar data points between the index patient and the control patient. This number corresponds to the number of controls (Age, Gender, etc.) chosen during the match control query creation process, corresponds to Match\_Strength in RPDR.

**Examples**

```
## Not run:
#Using defaults
d_mcm <- load_mcm(file = "test_Mcm.txt")

#Use sequential processing
d_mcm <- load_mcm(file = "test_Mcm.txt", nThread = 1)

#Use parallel processing and parse data in MRN_Type and MRN columns and keep all IDs
d_mcm <- load_mcm(file = "test_Mcm.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)
```

---

load_med	<i>Loads medication order detail into R.</i>
----------	--

---

## Description

Loads medication order detail information into the R environment, both Med and Mee files.

## Usage

```
load_med(
  file,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
  perc = 0.6,
  na = TRUE,
  identical = TRUE,
  nThread = parallel::detectCores() - 1,
  mrn_type = FALSE
)
```

## Arguments

file	string, full file path to Med.txt or Mee.txt
merge_id	string, column name to use to create <i>ID_MERGE</i> column used to merge different datasets. Defaults to <i>EPIC_PMRN</i> , as it is the preferred MRN in the RPDR system.
sep	string, divider between hospital ID and MRN. Defaults to <i>:</i> .
id_length	string, indicating whether to modify MRN length based-on required values <i>id_length = standard</i> , or to keep lengths as is <i>id_length = asis</i> . If <i>id_length = standard</i> then in case of <i>MGH</i> , <i>BWH</i> , <i>MCL</i> , <i>EMPI</i> and <i>PMRN</i> the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to <i>standard</i> .
perc	numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in <i>perc x 100%</i> of patients are kept.
na	boolean, whether to remove columns with only NA values. Defaults to <i>TRUE</i> .
identical	boolean, whether to remove columns with identical values. Defaults to <i>TRUE</i> .
nThread	integer, number of threads to use to load data.
mrn_type	boolean, should data in <i>MRN_Type</i> and <i>MRN</i> be parsed. Defaults to <i>FALSE</i> , as it is not advised to parse these for all data sources as it takes considerable time.

**Value**

data table, with medication order information.

**ID\_MERGE** numeric, defined IDs by *merge\_id*, used for merging later.

**ID\_med\_EMPI** string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information from *enc* datasource, corresponds to EMPI in RPDR. Data is formatted using `pretty_mrn()`.

**ID\_med\_PMRN** string, Epic medical record number. This value is unique across Epic instances within the Partners network from *enc* datasource, corresponds to EPIC\_PMRN in RPDR. Data is formatted using `pretty_mrn()`.

**ID\_med\_loc** string, if `mrn_type == TRUE`, then the data in *MRN\_Type* and *MRN* are parsed into IDs corresponding to locations (*loc*). Data is formatted using `pretty_mrn()`.

**med\_enc\_num** string, Unique identifier of the record/visit, displayed in the following format: Source System - Institution Number, corresponds to *Encounter\_number* in RPDR.

**time\_med** POSIXct, Completion status of the requested test/transfusion. Converted to POSIXct format, corresponds to *Medication\_Date* in RPDR.

**time\_med\_detail** string, To clarify when patients may have stopped taking a medication, this column provides the statuses of 'Listed' or 'Removed'. This is provided on pre-Epic (LMR) medication dates (1997-2017). The 'Listed' value denotes that a medication was on the patient's medication list on the date indicated. The 'Removed' value denotes that a medication was removed from a patient's medication list on the date indicated. Corresponds to *Medication\_Date\_Detail* in RPDR.

**med** string, Name of the medication. This may be appended with the source system in the case of OnCall and LMR medications, corresponds to *Medication* in RPDR.

**med\_code** string, Medication code associated with the "Code\_type" value, corresponds to *Code* in RPDR.

**med\_code\_type** string, Standardized classification system or custom source value used to identify the medication, corresponds to *Code\_Type* in RPDR.

**med\_quant** string, Number of units of the medication ordered, corresponds to *Quantity* in RPDR.

**med\_prov** string, Ordering provider for the medication, corresponds to *Provider* in RPDR.

**med\_clinic** string, Specific department/location where the medication was ordered or administered, corresponds to *Clinic* in RPDR.

**med\_hosp** string, Facility where the medication was ordered or administered, corresponds to *Hospital* in RPDR.

**med\_inpatient** string, Identifies whether the medication was ordered with an Inpatient or Outpatient indication, corresponds to *Inpatient\_Outpatient* in RPDR.

**med\_add\_info** string, Additional administration information about the medication, corresponds to *Additional\_Info* in RPDR.

**Examples**

```
## Not run:
#Using defaults
d_med <- load_med(file = "test_Med.txt")
```

```

#Use sequential processing
d_med <- load_med(file = "test_Med.txt", nThread = 1)

#Use parallel processing and parse data in MRN_Type and MRN columns and keep all IDs
d_mee <- load_med(file = "test_Mee.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)

```

---

load_mic	<i>Loads microbiology results into R.</i>
----------	---

---

## Description

Loads microbiology results into the R environment.

## Usage

```

load_mic(
  file,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
  perc = 0.6,
  na = TRUE,
  identical = TRUE,
  nThread = parallel::detectCores() - 1,
  mrn_type = FALSE,
  format_orig = FALSE
)

```

## Arguments

file	string, full file path to Mic.txt
merge_id	string, column name to use to create <i>ID_MERGE</i> column used to merge different datasets. Defaults to <i>EPIC_PMRN</i> , as it is the preferred MRN in the RPDR system.
sep	string, divider between hospital ID and MRN. Defaults to <code>:</code> .
id_length	string, indicating whether to modify MRN length based-on required values <i>id_length = standard</i> , or to keep lengths as is <i>id_length = asis</i> . If <i>id_length = standard</i> then in case of <i>MGH</i> , <i>BWH</i> , <i>MCL</i> , <i>EMPI</i> and <i>PMRN</i> the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to <i>standard</i> .
perc	numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in <i>perc x 100%</i> of patients are kept.
na	boolean, whether to remove columns with only NA values. Defaults to <i>TRUE</i> .

identical	boolean, whether to remove columns with identical values. Defaults to <i>TRUE</i> .
nThread	integer, number of threads to use to load data.
mrn_type	boolean, should data in <i>MRN_Type</i> and <i>MRN</i> be parsed. Defaults to <i>FALSE</i> , as it is not advised to parse these for all data sources as it takes considerable time.
format_orig	boolean, should report be returned in its original formatting or should white spaces used for formatting be removed. Defaults to <i>FALSE</i> .

## Value

data table, with microbiology information.

**ID\_MERGE** numeric, defined IDs by *merge\_id*, used for merging later.

**ID\_mic\_EMPI** string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information from *mic* datasource, corresponds to EMPI in RPDR. Data is formatted using *pretty\_mrn()*.

**ID\_mic\_PMRN** string, Epic medical record number. This value is unique across Epic instances within the Partners network from *mic* datasource, corresponds to EPIC\_PMRN in RPDR. Data is formatted using *pretty\_mrn()*.

**ID\_mic\_loc** string, if *mrn\_type* == *TRUE*, then the data in *MRN\_Type* and *MRN* are parsed into IDs corresponding to locations (*loc*). Data is formatted using *pretty\_mrn()*.

**time\_mic** POSIXct, Date when the specimen was received by the laboratory, corresponds to *Microbiology\_Date\_Time* in RPDR. Converted to POSIXct format.

**mic\_org\_code** string, Internal identifier for the organism used by the source system, corresponds to *Organism\_Code* in RPDR.

**mic\_org\_name** string, Name of the organism identified or tested, corresponds to *Organism\_Name* in RPDR.

**mic\_org\_text** string, Full narrative text of the test and results, including sensitivities, corresponds to *Organism\_Text* in RPDR.

**mic\_org\_comment** string, Free-text information about the organism or result, corresponds to *Organism\_Comment* in RPDR.

**mic\_test\_code** string, Internal identifier for the test used by the source system, corresponds to *Test\_Code* in RPDR.

**mic\_test\_name** string, Name of the assay to be performed, or the results of a culture, corresponds to *Test\_Name* in RPDR.

**mic\_test\_status** string, Status of the results, i.e. preliminary or final, corresponds to *Test\_Status* in RPDR.

**mic\_test\_comment** string, Free-text information about the test and results, corresponds to *Test\_Comments* in RPDR.

**mic\_spec** string, Type of specimen collected to perform the test, corresponds to *Specimen\_Type* in RPDR.

**mic\_spec\_txt** string, Free-text information about the specimen, its collection or its integrity, corresponds to *Specimen\_Comments* in RPDR.

**mic\_accession** string, Internal tracking number assigned to the specimen for identification in the microbiology lab, corresponds to *Microbiology\_Number* in RPDR.

**Examples**

```
## Not run:
#Using defaults
d_mic <- load_mic(file = "test_Mic.txt")

#Use sequential processing
d_mic <- load_mic(file = "test_Mic.txt", nThread = 1)

#Use parallel processing and parse data in MRN_Type and MRN columns and keep all IDs
d_mic <- load_mic(file = "test_Mic.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)
```

---

load_mrn	<i>Loads MRN data into R.</i>
----------	-------------------------------

---

**Description**

Loads patient identifiers for Partners institutions, including hospital-specific MRNs into the R environment.

**Usage**

```
load_mrn(
  file,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
  perc = 0.6,
  na = TRUE,
  identical = TRUE,
  nThread = parallel::detectCores() - 1,
  mrn_type = FALSE
)
```

**Arguments**

file	string, full file path to Mrn.txt.
merge_id	string, column name to use to create <i>ID_MERGE</i> column used to merge different datasets. Defaults to <i>EPIC_PMRN</i> , as it is the preferred MRN in the RPDR system.
sep	string, divider between hospital ID and MRN. Defaults to <code>:</code> .
id_length	string, indicating whether to modify MRN length based-on required values <i>id_length</i> = <i>standard</i> , or to keep lengths as is <i>id_length</i> = <i>asis</i> . If <i>id_length</i> = <i>standard</i> then in case of <i>MGH</i> , <i>BWH</i> , <i>MCL</i> , <i>EMPI</i> and <i>PMRN</i> the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to <i>standard</i> .

perc	numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in <i>perc x 100%</i> of patients are kept. Not used for loading mrn data.
na	boolean, whether to remove columns with only NA values. Defaults to <i>TRUE</i> .
identical	boolean, whether to remove columns with identical values. Defaults to <i>TRUE</i> .
nThread	integer, number of threads to use to load data.
mrn_type	boolean, should data in <i>MRN_Type</i> and <i>MRN</i> be parsed. Defaults to <i>FALSE</i> , as it is not advised to parse these for all data sources as it takes considerable time.

### Value

data table, with MRN data.

**ID\_MERGE** numeric, defined IDs by *merge\_id*, used for merging later.

**ID\_mrn\_INCOMING** string, Patient identifier, usually the EMPI, corresponds to IncomingId in RPDR. Data is formatted using *pretty\_mrn()*.

**ID\_mrn\_INCOMING\_SITE** string, Source of identifier, e.g. EMP for Enterprise Master Patient Index, MGH for Mass General Hospital, corresponds to IncomingSite in RPDR.

**ID\_mrn\_PMRN** string, Epic medical record number. This value is unique across Epic instances within the Partners network, corresponds to EPIC\_PMRN in RPDR. Data is formatted using *pretty\_mrn()*.

**ID\_mrn\_EMPI** string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information, corresponds to Enterprise\_Master\_Patient\_Index in RPDR. Data is formatted using *pretty\_mrn()*.

**ID\_mrn\_MGH** string, Unique Medical Record Number for Mass General Hospital, corresponds to MGH\_MRN in RPDR. Data is formatted using *pretty\_mrn()*.

**ID\_mrn\_BWH** string, Unique Medical Record Number for Brigham and Women's Hospital, corresponds to BWH\_MRN in RPDR. Data is formatted using *pretty\_mrn()*.

**ID\_mrn\_FH** string, Unique Medical Record Number for Faulkner Hospital, corresponds to FH\_MRN in RPDR. Data is formatted using *pretty\_mrn()*.

**ID\_mrn\_SRH** string, Unique Medical Record Number for Spaulding Rehabilitation Hospital, corresponds to SRH\_MRN in RPDR. Data is formatted using *pretty\_mrn()*.

**ID\_mrn\_NWH** string, Unique Medical Record Number for Newton-Wellesley Hospital, corresponds to NWH\_MRN in RPDR. Data is formatted using *pretty\_mrn()*.

**ID\_mrn\_NSMC** string, Unique Medical Record Number for North Shore Medical Center, corresponds to NSMC\_MRN in RPDR. Data is formatted using *pretty\_mrn()*.

**ID\_mrn\_MCL** string, Unique Medical Record Number for McLean Hospital, corresponds to MCL\_MRN in RPDR. Data is formatted using *pretty\_mrn()*.

**ID\_mrn\_MEE** string, Unique Medical Record Number for Mass Eye and Ear, corresponds to MEE\_MRN in RPDR. Data is formatted using *pretty\_mrn()*.

**ID\_mrn\_DFC** string, Unique Medical Record Number for Dana Farber Cancer center, corresponds to DFC\_MRN in RPDR. Data is formatted using *pretty\_mrn()*.

**ID\_mrn\_WDH** string, Unique Medical Record Number for Wentworth-Douglass Hospital, corresponds to WDH\_MRN in RPDR. Data is formatted using *pretty\_mrn()*.

**ID\_mrn\_STATUS** string, Status of the record, corresponds to Status in RPDR.

## Examples

```
## Not run:
#Using defaults
d_mrn <- load_mrn(file = "test_Mrn.txt")

#Use sequential processing
d_mrn <- load_mrn(file = "test_Mrn.txt", nThread = 1)

#Use parallel processing and parse data in MRN_Type and MRN columns and keep all IDs
d_mrn <- load_mrn(file = "test_Mrn.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)
```

---

load\_notes

*Loads note documents into R.*

---

## Description

Loads documents information into the R environment, which are:

**Cardiology:** "car"

**Discharge:** "dis"

**Endoscopy:** "end"

**History & Physical:** "hnp"

**Operative:** "opn"

**Pathology:** "pat"

**Progress:** "prg"

**Pulmonary:** "pul"

**Radiology:** "rad"

**Visit:** "vis"

## Usage

```
load_notes(
  file,
  type,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
  perc = 0.6,
  na = TRUE,
  identical = TRUE,
  nThread = parallel::detectCores() - 1,
  mrn_type = FALSE,
  load_report = TRUE,
  format_orig = FALSE
)
```

**Arguments**

file	string, full file path to given type of note i.e. Hnp.txt.
type	string, the type of note to be loaded. May be on of: "car", "dis", "end", "hnp", "opn", "pat", "prg", "pul", "rad" or "vis".
merge_id	string, column name to use to create <i>ID_MERGE</i> column used to merge different datasets. Defaults to <i>EPIC_PMRN</i> , as it is the preferred MRN in the RPDR system.
sep	string, divider between hospital ID and MRN. Defaults to <code>.</code> .
id_length	string, indicating whether to modify MRN length based-on required values <i>id_length</i> = <i>standard</i> , or to keep lengths as is <i>id_length</i> = <i>asis</i> . If <i>id_length</i> = <i>standard</i> then in case of <i>MGH</i> , <i>BWH</i> , <i>MCL</i> , <i>EMPI</i> and <i>PMRN</i> the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to <i>standard</i> .
perc	numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in <i>perc</i> x 100% of patients are kept.
na	boolean, whether to remove columns with only NA values. Defaults to <i>TRUE</i> .
identical	boolean, whether to remove columns with identical values. Defaults to <i>TRUE</i> .
nThread	integer, number of threads to use to load data.
mrn_type	boolean, should data in <i>MRN_Type</i> and <i>MRN</i> be parsed. Defaults to <i>FALSE</i> , as it is not advised to parse these for all data sources as it takes considerable time.
load_report	boolean, should the report text be returned in the data table. Defaults to <i>TRUE</i> . However, be aware that some notes may take up more memory than available on the machine.
format_orig	boolean, should report be returned in its original formatting or should white spaces used for formatting be removed. Defaults to <i>FALSE</i> .

**Value**

data table, with notes information. *abc* stands for the three letter abbreviation of the given type of note.

**ID\_MERGE** numeric, defined IDs by *merge\_id*, used for merging later.

**ID\_abc\_EMPI** string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information from *abc* datasource, corresponds to EMPI in RPDR. Data is formatted using `pretty_mrn()`.

**ID\_abc\_PMRN** string, Epic medical record number. This value is unique across Epic instances within the Partners network from *abc* datasource, corresponds to EPIC\_PMRN in RPDR. Data is formatted using `pretty_mrn()`.

**ID\_abc\_loc** string, if *mrn\_type* == *TRUE*, then the data in *MRN\_Type* and *MRN* are parsed into IDs corresponding to locations (*loc*). Data is formatted using `pretty_mrn()`.

**abc\_rep\_num** string, Source-specific identifier used to reference the report, corresponds to Report\_Number in RPDR.

**time\_abc** POSIXct, Date when the report was filed, corresponds to Report\_Date\_Time in RPDR. Converted to POSIXct format.

- abc\_rep\_desc** string, Type of report or procedure documented in the report, corresponds to Report\_Description in RPDR.
- abc\_rep\_status** string, Completion status of the note/report, corresponds to Report\_Status in RPDR.
- abc\_rep\_type** string, See specification in RPDR data dictionary, corresponds to Report\_Type in RPDR.
- abc\_rep\_txt** string, Full narrative text contained in the note/report, corresponds to Report\_Text in RPDR. Only provided if *load\_report* is TRUE.

### Examples

```
## Not run:
#Using defaults
d_hnp <- load_notes(file = "test_Hnp.txt", type = "hnp")

#Use sequential processing
d_hnp <- load_notes(file = "test_Hnp.txt", type = "hnp", nThread = 1, format_orig = TRUE)

#Use parallel processing and parse data in MRN_Type and MRN columns and keep all IDs
d_hnp <- load_notes(file = "test_Hnp.txt", type = "hnp", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)
```

---

load\_phy

*Loads helath history information into R.*

---

### Description

Loads vital signs, social history, immunizations, and various other health history details into the R environment.

### Usage

```
load_phy(
  file,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
  perc = 0.6,
  na = TRUE,
  identical = TRUE,
  nThread = parallel::detectCores() - 1,
  mrn_type = FALSE
)
```

**Arguments**

file	string, full file path to Phy.txt.
merge_id	string, column name to use to create <i>ID_MERGE</i> column used to merge different datasets. Defaults to <i>EPIC_PMRN</i> , as it is the preferred MRN in the RPDR system.
sep	string, divider between hospital ID and MRN. Defaults to <code>:</code> .
id_length	string, indicating whether to modify MRN length based-on required values <i>id_length</i> = <i>standard</i> , or to keep lengths as is <i>id_length</i> = <i>asis</i> . If <i>id_length</i> = <i>standard</i> then in case of <i>MGH</i> , <i>BWH</i> , <i>MCL</i> , <i>EMPI</i> and <i>PMRN</i> the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to <i>standard</i> .
perc	numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in <i>perc</i> x 100% of patients are kept.
na	boolean, whether to remove columns with only NA values. Defaults to <i>TRUE</i> .
identical	boolean, whether to remove columns with identical values. Defaults to <i>TRUE</i> .
nThread	integer, number of threads to use to load data.
mrn_type	boolean, should data in <i>MRN_Type</i> and <i>MRN</i> be parsed. Defaults to <i>FALSE</i> , as it is not advised to parse these for all data sources as it takes considerable time.

**Value**

data table, with health history information.

**ID\_MERGE** numeric, defined IDs by *merge\_id*, used for merging later.

**ID\_phy\_EMPI** string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information from *phy* datasource, corresponds to EMPI in RPDR. Data is formatted using `pretty_mrn()`.

**ID\_phy\_PMRN** string, Epic medical record number. This value is unique across Epic instances within the Partners network from *phy* datasource, corresponds to EPIC\_PMRN in RPDR. Data is formatted using `pretty_mrn()`.

**ID\_phy\_loc** string, if *mrn\_type* == *TRUE*, then the data in *MRN\_Type* and *MRN* are parsed into IDs corresponding to locations (*loc*). Data is formatted using `pretty_mrn()`.

**time\_phy** POSIXct, Date when the diagnosis was noted, corresponds to Date in RPDR. Converted to POSIXct format.

**phy\_name** string, Type of clinical value/observation recorded, corresponds to Concept\_Name in RPDR.

**phy\_code** string, Source-specific identifier for the specific type of clinical observation, corresponds to Code in RPDR.

**phy\_code\_type** string, Source system for the value, corresponds to Code\_type in RPDR.

**phy\_result** string, Value associated with the clinical observation. Note: BMI results are calculated internally in the RPDR, corresponds to Results in RPDR.

**phy\_unit** string, Units associated with the clinical observation, corresponds to Units in RPDR.

**phy\_provider** string, Provider of record for the encounter where the observation was recorded, corresponds to Providers in RPDR.

**phy\_clinic** string, Specific department/location where the patient observation was recorded, corresponds to Clinic in RPDR.

**phy\_hosp** string, Facility where the observation was recorded, corresponds to Hospital in RPDR.

**phy\_inpatient** string, Classifies the type of encounter where the observation was entered, corresponds to Inpatient\_Outpatient in RPDR.

**phy\_enc\_num** string, Unique identifier of the record/visit. This values includes the source system and a unique identifier within the source system, corresponds to Encounter\_number in RPDR.

### Examples

```
## Not run:
#Using defaults
d_phy <- load_phy(file = "test_Phy.txt")

#Use sequential processing
d_phy <- load_phy(file = "test_Phy.txt", nThread = 1)

#Use parallel processing and parse data in MRN_Type and MRN columns and keep all IDs
d_phy <- load_phy(file = "test_Phy.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)
```

---

load\_prc

*Loads procedures into R.*

---

### Description

Loads Clinical procedure information into the R environment, both Prc and Pec files.

### Usage

```
load_prc(
  file,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
  perc = 0.6,
  na = TRUE,
  identical = TRUE,
  nThread = parallel::detectCores() - 1,
  mrn_type = FALSE
)
```

**Arguments**

file	string, full file path to Prc.txt or Pec.txt.
merge_id	string, column name to use to create <i>ID_MERGE</i> column used to merge different datasets. Defaults to <i>EPIC_PMRN</i> , as it is the preferred MRN in the RPDR system.
sep	string, divider between hospital ID and MRN. Defaults to <code>:</code> .
id_length	string, indicating whether to modify MRN length based-on required values <i>id_length</i> = <i>standard</i> , or to keep lengths as is <i>id_length</i> = <i>asis</i> . If <i>id_length</i> = <i>standard</i> then in case of <i>MGH</i> , <i>BWH</i> , <i>MCL</i> , <i>EMPI</i> and <i>PMRN</i> the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to <i>standard</i> .
perc	numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in <i>perc</i> x 100% of patients are kept.
na	boolean, whether to remove columns with only NA values. Defaults to <i>TRUE</i> .
identical	boolean, whether to remove columns with identical values. Defaults to <i>TRUE</i> .
nThread	integer, number of threads to use to load data.
mrn_type	boolean, should data in <i>MRN_Type</i> and <i>MRN</i> be parsed. Defaults to <i>FALSE</i> , as it is not advised to parse these for all data sources as it takes considerable time.

**Value**

data table, with procedural information.

**ID\_MERGE** numeric, defined IDs by *merge\_id*, used for merging later.

**ID\_prc\_EMPI** string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information from *prc* datasource, corresponds to EMPI in RPDR. Data is formatted using `pretty_mrn()`.

**ID\_prc\_PMRN** string, Epic medical record number. This value is unique across Epic instances within the Partners network from *prc* datasource, corresponds to EPIC\_PMRN in RPDR. Data is formatted using `pretty_mrn()`.

**ID\_prc\_loc** string, if *mrn\_type* == *TRUE*, then the data in *MRN\_Type* and *MRN* are parsed into IDs corresponding to locations (*loc*). Data is formatted using `pretty_mrn()`.

**time\_prc** POSIXct, Date when the procedure was performed, corresponds to Date in RPDR. Converted to POSIXct format.

**prc\_name** string, Name of the procedure or operation performed, corresponds to Procedure\_Name in RPDR.

**prc\_code** string, Procedure code associated with the "Code\_type" value, corresponds to Code in RPDR.

**prc\_code\_type** string, Standardized classification system or custom source value associated with the procedure code, corresponds to Code\_type in RPDR.

**prc\_flag** string, Qualifier for the diagnosis, corresponds to Procedure\_Flag in RPDR.

**prc\_quantity** string, Number of the procedures that were ordered for this record, corresponds to Quantity in RPDR.

- prc\_provider** string, Provider identifies the health care clinician performing the procedure, corresponds to Provider in RPDR.
- prc\_clinic** string, Specific department/location where the procedure was ordered or performed, corresponds to Clinic in RPDR.
- prc\_hosp** string, Facility where the procedure was ordered or performed, corresponds to Hospital in RPDR.
- prc\_inpatient** string, classifies the type of encounter where the procedure was performed or ordered.
- prc\_enc\_num** string, Unique identifier of the record/visit, displayed in the following format: Source System - Institution Number, corresponds to Encounter\_number in RPDR.

### Examples

```
## Not run:
#Using defaults
d_prv <- load_prv(file = "test_Prv.txt")

#Use sequential processing
d_prv <- load_prv(file = "test_Prv.txt", nThread = 1)

#Use parallel processing and parse data in MRN_Type and MRN columns and keep all IDs
d_pec <- load_prv(file = "test_Pec.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)
```

---

load\_prv

*Loads providers information into R.*

---

### Description

Loads providers information into the R environment.

### Usage

```
load_prv(
  file,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
  perc = 0.6,
  na = TRUE,
  identical = TRUE,
  nThread = parallel::detectCores() - 1,
  mrn_type = TRUE
)
```

**Arguments**

file	string, full file path to Prv.txt.
merge_id	string, column name to use to create <i>ID_MERGE</i> column used to merge different datasets. Defaults to <i>EPIC_PMRN</i> , as it is the preferred MRN in the RPDR system.
sep	string, divider between hospital ID and MRN. Defaults to <code>.</code>
id_length	string, indicating whether to modify MRN length based-on required values <i>id_length = standard</i> , or to keep lengths as is <i>id_length = asis</i> . If <i>id_length = standard</i> then in case of <i>MGH, BWH, MCL, EMPI and PMRN</i> the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to <i>standard</i> .
perc	numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in <i>perc x 100%</i> of patients are kept.
na	boolean, whether to remove columns with only NA values. Defaults to <i>TRUE</i> .
identical	boolean, whether to remove columns with identical values. Defaults to <i>TRUE</i> .
nThread	integer, number of threads to use to load data.
mrn_type	boolean, should data in <i>MRN_Type</i> and <i>MRN</i> be parsed. Defaults to <i>TURE</i> only for Con.txt, as it is not advised to parse these for all data sources as it takes considerable time.

**Value**

data table, with provider information data.

**ID\_MERGE** numeric, defined IDs by *merge\_id*, used for merging later.

**ID\_con\_EMPI** string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information from *con* datasource, corresponds to EMPI in RPDR. Data is formatted using *pretty\_mrn()*.

**ID\_con\_PMRN** string, Epic medical record number. This value is unique across Epic instances within the Partners network from *con*datasource, corresponds to EPIC\_PMRN in RPDR. Data is formatted using *pretty\_mrn()*.

**ID\_con\_loc** string, if *mrn\_type == TRUE*, then the data in *MRN\_Type* and *MRN* are parsed into IDs corresponding to locations (*loc*). Data is formatted using *pretty\_mrn()*.

**time\_prv\_last\_seen** POSIXct, Date when the patient was last seen by the provider, corresponds to *Last\_Seen\_Date* in RPDR.

**prv\_name** string, Full name of the provider, corresponds to *Provider\_Name* in RPDR.

**prv\_rank** string, Provides a quantitative value of provider's level of interaction with the patient. This is calculated using the number of CPT codes for face-to-face visits that the provider has billed for in relation to the patient, corresponds to *Provider\_Rank* in RPDR.

**prv\_ID** string, Identification code for the provider, including the source institution, corresponds to *Provider\_ID* in RPDR.

**prv\_ID\_CMP** string, Corporate Provider Master ID. This is the unique identifier for a provider across the MGB network, corresponds to *CPM\_Id* in RPDR.

**prv\_spec** string, Comma-delimited list of the provider's specialties, corresponds to Specialties in RPDR.

**prv\_pcp** string, Available for BWH and MGH PCPs only. Flag indicating whether the provider is listed as the patient's Primary Care Physician, corresponds to Is\_PCP in RPDR.

**prv\_dep** string, Provider's department, corresponds to Enterprise\_service in RPDR.

**prv\_address1** string, Address of the provider's primary practice, corresponds to Address\_1 in RPDR.

**prv\_address2** string, Additional address information, corresponds to Address\_2 in RPDR.

**prv\_city** string, City of the provider's primary practice, corresponds to City in RPDR.

**prv\_state** string, State of the provider's primary practice, corresponds to State in RPDR.

**prv\_zip** string, Mailing zip code of provider's primary practice, corresponds to Zip in RPDR.

**prv\_phone** string, Telephone number of the provider's primary practice, corresponds to Phone\_Ext in RPDR.

**prv\_fax** string, Fax number of the provider's primary practice, corresponds to Fax in RPDR.

**prv\_email** string, Primary e-mail address for the provider, corresponds to Email in RPDR.

### Examples

```
## Not run:
#Using defaults
d_prv <- load_prv(file = "test_Priv.txt")

#Use sequential processing
d_prv <- load_prv(file = "test_Priv.txt", nThread = 1)

#Use parallel processing and parse data in
#MRN_Type and MRN columns (default in load_con) and keep all IDs
d_prv <- load_prv(file = "test_Priv.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)
```

---

load\_ptd

*Loads patient data information into R.*

---

### Description

Loads patient data information into the R environment.

### Usage

```
load_ptd(
  file,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
```

```

perc = 0.6,
na = TRUE,
identical = TRUE,
nThread = parallel::detectCores() - 1,
mrn_type = FALSE
)

```

### Arguments

file	string, full file path to Ptd.txt.
merge_id	string, column name to use to create <i>ID_MERGE</i> column used to merge different datasets. Defaults to <i>EPIC_PMRN</i> , as it is the preferred MRN in the RPDR system.
sep	string, divider between hospital ID and MRN. Defaults to <code>.</code> .
id_length	string, indicating whether to modify MRN length based-on required values <i>id_length = standard</i> , or to keep lengths as is <i>id_length = asis</i> . If <i>id_length = standard</i> then in case of <i>MGH</i> , <i>BWH</i> , <i>MCL</i> , <i>EMPI</i> and <i>PMRN</i> the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to <i>standard</i> .
perc	numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in <i>perc x 100%</i> of patients are kept.
na	boolean, whether to remove columns with only NA values. Defaults to <i>TRUE</i> .
identical	boolean, whether to remove columns with identical values. Defaults to <i>TRUE</i> .
nThread	integer, number of threads to use to load data.
mrn_type	boolean, should data in <i>MRN_Type</i> and <i>MRN</i> be parsed. Defaults to <i>FALSE</i> , as it is not advised to parse these for all data sources as it takes considerable time.

### Value

data table, with patient data information information.

**ID\_MERGE** numeric, defined IDs by *merge\_id*, used for merging later.

**ID\_ptd\_EMPI** string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information from *ptd* datasource, corresponds to EMPI in RPDR. Data is formatted using *pretty\_mrn()*.

**ID\_ptd\_PMRN** string, Epic medical record number. This value is unique across Epic instances within the Partners network from *ptd* datasource, corresponds to EPIC\_PMRN in RPDR. Data is formatted using *pretty\_mrn()*.

**ID\_ptd\_loc** string, if *mrn\_type == TRUE*, then the data in *MRN\_Type* and *MRN* are parsed into IDs corresponding to locations (*loc*). Data is formatted using *pretty\_mrn()*.

**time\_ptd\_start** POSIXct, Date item was initiated in the record, corresponds to *Start\_Date* in RPDR. Converted to POSIXct format.

**time\_ptd\_end** POSIXct, Date item was finalized in the record, corresponds to *End\_Date* in RPDR. Converted to POSIXct format.

**ptd\_desc** string, Name of the item being reported, corresponds to *Description* in RPDR.

- ptd\_result** string, Result of the item being reported, corresponds to Result in RPDR.
- ptd\_type** string, Describes the type of data being reported, corresponds to Patient\_Data\_Type in RPDR.
- ptd\_enc\_num** string, Unique identifier of the record/visit. This values includes the source system and a unique identifier within the source system, corresponds to Encounter\_number in RPDR.

### Examples

```
## Not run:
#Using defaults
d_ptd <- load_ptd(file = "test_Phy.txt")

#Use sequential processing
d_ptd <- load_ptd(file = "test_Phy.txt", nThread = 1)

#Use parallel processing and parse data in MRN_Type and MRN columns and keep all IDs
d_ptd <- load_ptd(file = "test_Phy.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)
```

---

load_rdt	<i>Loads radiology procedures data into R.</i>
----------	--

---

### Description

Loads radiology procedures information into the R environment.

### Usage

```
load_rdt(
  file,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
  perc = 0.6,
  na = TRUE,
  identical = TRUE,
  nThread = parallel::detectCores() - 1,
  mrn_type = FALSE
)
```

### Arguments

file	string, full file path to Rdt.txt.
merge_id	string, column name to use to create <i>ID_MERGE</i> column used to merge different datasets. Defaults to <i>EPIC_PMRN</i> , as it is the preferred MRN in the RPDR system.

sep	string, divider between hospital ID and MRN. Defaults to <code>.</code> .
id_length	string, indicating whether to modify MRN length based-on required values <code>id_length = standard</code> , or to keep lengths as is <code>id_length = asis</code> . If <code>id_length = standard</code> then in case of <i>MGH</i> , <i>BWH</i> , <i>MCL</i> , <i>EMPI</i> and <i>PMRN</i> the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to <i>standard</i> .
perc	numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in <i>perc x 100%</i> of patients are kept.
na	boolean, whether to remove columns with only NA values. Defaults to <i>TRUE</i> .
identical	boolean, whether to remove columns with identical values. Defaults to <i>TRUE</i> .
nThread	integer, number of threads to use to load data.
mrn_type	boolean, should data in <i>MRN_Type</i> and <i>MRN</i> be parsed. Defaults to <i>FALSE</i> , as it is not advised to parse these for all data sources as it takes considerable time.

### Value

data table, with radiological exam information.

**ID\_MERGE** numeric, defined IDs by *merge\_id*, used for merging later.

**ID\_rdt\_EMPI** string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information from *rdt* datasource, corresponds to EMPI in RPDR. Data is formatted using `pretty_mrn()`.

**ID\_rdt\_PMRN** string, Epic medical record number. This value is unique across Epic instances within the Partners network from *rdt* datasource, corresponds to EPIC\_PMRN in RPDR. Data is formatted using `pretty_mrn()`.

**ID\_rdt\_loc** string, if `mrn_type == TRUE`, then the data in *MRN\_Type* and *MRN* are parsed into IDs corresponding to locations (*loc*). Data is formatted using `pretty_mrn()`.

**time\_rdt\_exam** POSIXct, Date of the radiology exam, corresponds to Date in RPDR. Converted to POSIXct format.

**rdt\_mode** string, Modality of the exam, corresponds to Mode in RPDR.

**rdt\_group** string, Higher-level grouping concept used to consolidate similar procedures across hospitals, corresponds to Group in RPDR.

**rdt\_test\_code** string, Internal identifier for the procedure used by the source system, corresponds to Test\_Code in RPDR.

**rdt\_test\_desc** string, Full name of the exam/study performed, corresponds to Test\_Description in RPDR.

**rdt\_accession** string, Identifier assigned to the report or procedure for Radiology tracking purposes, corresponds to Accession\_Number in RPDR.

**rdt\_provider** string, Ordering or authorizing provider for the study, corresponds to Provider in RPDR.

**rdt\_clinic** string, Specific department/location where the procedure was ordered or performed, corresponds to Clinic in RPDR.

**rdt\_hosp** string, Facility where the order was entered, corresponds to Hospital in RPDR.

**rdt\_inpatient** string, Classifies the type of encounter where the procedure was performed, corresponds to Inpatient\_Outpatient in RPDR.

**Examples**

```
## Not run:
#Using defaults
d_rdt <- load_rdt(file = "test_Rdt.txt")

#Use sequential processing
d_rdt <- load_rdt(file = "test_Rdt.txt", nThread = 1)

#Use parallel processing and parse data in MRN_Type and MRN columns and keep all IDs
d_rdt <- load_rdt(file = "test_Rdt.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)
```

---

load_rfv	<i>Loads reason for visit data into R.</i>
----------	--

---

**Description**

Loads reason for visit information into the R environment.

**Usage**

```
load_rfv(
  file,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
  perc = 0.6,
  na = TRUE,
  identical = TRUE,
  nThread = parallel::detectCores() - 1,
  mrn_type = FALSE
)
```

**Arguments**

file	string, full file path to Rfv.txt.
merge_id	string, column name to use to create <i>ID_MERGE</i> column used to merge different datasets. Defaults to <i>EPIC_PMRN</i> , as it is the preferred MRN in the RPDR system.
sep	string, divider between hospital ID and MRN. Defaults to <code>:</code> .
id_length	string, indicating whether to modify MRN length based-on required values <i>id_length</i> = <i>standard</i> , or to keep lengths as is <i>id_length</i> = <i>asis</i> . If <i>id_length</i> = <i>standard</i> then in case of <i>MGH</i> , <i>BWH</i> , <i>MCL</i> , <i>EMPI</i> and <i>PMRN</i> the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to <i>standard</i> .

<code>perc</code>	numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in <i>perc x 100%</i> of patients are kept.
<code>na</code>	boolean, whether to remove columns with only NA values. Defaults to <i>TRUE</i> .
<code>identical</code>	boolean, whether to remove columns with identical values. Defaults to <i>TRUE</i> .
<code>nThread</code>	integer, number of threads to use to load data.
<code>mrn_type</code>	boolean, should data in <i>MRN_Type</i> and <i>MRN</i> be parsed. Defaults to <i>FALSE</i> , as it is not advised to parse these for all data sources as it takes considerable time.

### Value

data table, with reason for visit information.

**ID\_MERGE** numeric, defined IDs by *merge\_id*, used for merging later.

**ID\_rfv\_EMPI** string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information from *dia* datasource, corresponds to EMPI in RPDR. Data is formatted using `pretty_mrn()`.

**ID\_rfv\_PMRN** string, Epic medical record number. This value is unique across Epic instances within the Partners network from *rfv* datasource, corresponds to EPIC\_PMRN in RPDR. Data is formatted using `pretty_mrn()`.

**ID\_rfv\_loc** string, if `mrn_type == TRUE`, then the data in *MRN\_Type* and *MRN* are parsed into IDs corresponding to locations (*loc*). Data is formatted using `pretty_mrn()`.

**time\_rfv\_start** POSIXct, Start date of the encounter, corresponds to *Start\_Date* in RPDR. Converted to POSIXct format.

**time\_rfv\_end** POSIXct, End date of the encounter, corresponds to *End\_Date* in RPDR. Converted to POSIXct format.

**rfv\_provider** string, Primary provider for the encounter, corresponds to *Provider* in RPDR.

**rfv\_hosp** string, Facility where the encounter occurred, corresponds to *Hospital* in RPDR.

**rfv\_clinic** string, Specific department/location where the patient encounter took place, corresponds to *Clinic* in RPDR.

**rfv\_chief\_complaint** string, Description of the chief complaint/reason for visit, corresponds to *Chief\_Complaint* in RPDR.

**rfv\_concept\_id** string, Epic identifier for the chief complaint/reason for visit, corresponds to *Concept\_id* in RPDR.

**rfv\_comment** string, Free-text comments regarding the chief complain/reason for visit, corresponds to *Comments* in RPDR.

**rfv\_enc\_num** string, Unique identifier of the record/visit. This values includes the source system, hospital, and a unique identifier within the source system, corresponds to *Encounter\_number* in RPDR.

### Examples

```
## Not run:
#Using defaults
d_rfv <- load_rfv(file = "test_Rfv.txt")
```

```

#Use sequential processing
d_rfv <- load_rfv(file = "test_Rfv.txt", nThread = 1)

#Use parallel processing and parse data in MRN_Type and MRN columns and keep all IDs
d_rfv <- load_rfv(file = "test_Rfv.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)

```

---

load_trn	<i>Loads transfusion results into R.</i>
----------	--

---

## Description

Loads transfusion results into the R environment.

## Usage

```

load_trn(
  file,
  merge_id = "EMPI",
  sep = ":",
  id_length = "standard",
  perc = 0.6,
  na = TRUE,
  identical = TRUE,
  nThread = parallel::detectCores() - 1,
  mrn_type = FALSE
)

```

## Arguments

file	string, full file path to Trn.txt
merge_id	string, column name to use to create <i>ID_MERGE</i> column used to merge different datasets. Defaults to <i>EPIC_PMRN</i> , as it is the preferred MRN in the RPDR system.
sep	string, divider between hospital ID and MRN. Defaults to <code>:</code> .
id_length	string, indicating whether to modify MRN length based-on required values <i>id_length</i> = <i>standard</i> , or to keep lengths as is <i>id_length</i> = <i>asis</i> . If <i>id_length</i> = <i>standard</i> then in case of <i>MGH</i> , <i>BWH</i> , <i>MCL</i> , <i>EMPI</i> and <i>PMRN</i> the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to <i>standard</i> .
perc	numeric, a number between 0-1 indicating which parsed ID columns to keep. Data present in <i>perc</i> x 100% of patients are kept.
na	boolean, whether to remove columns with only NA values. Defaults to <i>TRUE</i> .
identical	boolean, whether to remove columns with identical values. Defaults to <i>TRUE</i> .
nThread	integer, number of threads to use to load data.
mrn_type	boolean, should data in <i>MRN_Type</i> and <i>MRN</i> be parsed. Defaults to <i>FALSE</i> , as it is not advised to parse these for all data sources as it takes considerable time.

**Value**

data table, with transfusion information.

**ID\_MERGE** numeric, defined IDs by *merge\_id*, used for merging later.

**ID\_trn\_EMPI** string, Unique Partners-wide identifier assigned to the patient used to consolidate patient information from *trn* datasource, corresponds to EMPI in RPDR. Data is formatted using `pretty_mrn()`.

**ID\_trn\_PMRN** string, Epic medical record number. This value is unique across Epic instances within the Partners network from *trn* datasource, corresponds to EPIC\_PMRN in RPDR. Data is formatted using `pretty_mrn()`.

**ID\_trn\_loc** string, if `mrn_type == TRUE`, then the data in *MRN\_Type* and *MRN* are parsed into IDs corresponding to locations (*loc*). Data is formatted using `pretty_mrn()`.

**time\_trn** POSIXct, Date when the transfusion was administered or test was performed, corresponds to *Transaction\_Date\_Time* in RPDR. Converted to POSIXct format.

**trn\_descript** string, The type of procedure or product administered, corresponds to *Test\_Description* in RPDR.

**trn\_result** string, Results of the test or transaction/lot number of transfusion, corresponds to *Results* in RPDR.

**trn\_result\_abn** string, Denotes an abnormal finding or value, corresponds to *Abnormal\_Flag* in RPDR.

**trn\_comment** string, Free-text comments about the status of the test/transfusion, corresponds to *Comments* in RPDR.

**trn\_status** string, Completion status of the requested test/transfusion, corresponds to *Status\_Flag* in RPDR.

**trn\_accession** string, Identifier assigned to the test/transfusion for tracking purposes by the blood bank, corresponds to *Accession* in RPDR.

**Examples**

```
## Not run:
#Using defaults
d_trn <- load_trn(file = "test_Trn.txt")

#Use sequential processing
d_trn <- load_trn(file = "test_Trn.txt", nThread = 1)

#Use parallel processing and parse data in MRN_Type and MRN columns and keep all IDs
d_trn <- load_trn(file = "test_Trn.txt", nThread = 20, mrn_type = TRUE, perc = 1)

## End(Not run)
```

---

pretty_mrn	<i>Converts MRN integer to string compatible with RPDR.</i>
------------	---

---

### Description

Adds or removes zeros from integers to comply with MRN code standards for given institution and adds institution prefix.

### Usage

```
pretty_mrn(v, prefix = "MGH", sep = ":", id_length = "standard", nThread = 1)
```

### Arguments

v	vector, integer or sting vector with MRNs.
prefix	string or vector, hospital ID from where the MRNs are from. Defaults to <i>MGH</i> . If a vector is provided then it must be the same length as <i>v</i> . This allows to potentially use different prefixes for different IDs using the same vector of values.
sep	string, divider between hospital ID and MRN. Defaults to <i>:</i> .
id_length	string, indicating whether to modify MRN length based-on required values <i>id_length = standard</i> , or to keep lengths as is <i>id_length = asis</i> . If <i>id_length = standard</i> then in case of <i>MGH</i> , <i>BWH</i> , <i>MCL</i> , <i>EMPI</i> and <i>PMRN</i> the length of the MRNs are corrected accordingly by adding zeros, or removing numeral from the beginning. In other cases the lengths are unchanged. Defaults to <i>standard</i> .
nThread	integer, number of threads to use by <i>dopar</i> for parallelization. If it is set to 1, then no parallel backends are created and the function is executed sequentially.

### Value

vector, with characters formatted to specified lengths. If length of the ID does not match the required length, then leading zeros are added to the ID. If the ID is longer then the required length, then numerals from the beginning of the ID are cut off until it is the required length.

### Examples

```
## Not run:
mrns <- sample(1e4:1e7, size = 10) #Simulate MRNs

#MGH format
pretty_mrn(v = mrns, prefix = "MGH")

#BWH format
pretty_mrn(v = mrns, prefix = "BWH")

#Multiple sources using space as a separator
pretty_mrn(v = mrns[1:3], prefix = c("MGH", "BWH", "EMPI"), sep = " ")
```

```
#Keeping the length of the IDs despite not adhering to the requirements
pretty_mrn(v = mrns, prefix = "EMPI", id_length = "asis")

## End(Not run)
```

---

```
pretty_numbers          Converts numerical codes to universal format specified by length.
```

---

### Description

Creates numerical strings with given lengths by removing additional characters from the back and adding leading zeros if necessary.

### Usage

```
pretty_numbers(v, length_final = 5, remove_from_back = 4)
```

### Arguments

`v` vector, integer or sting vector with numerical values.

`length_final` numeric, the length of the final string. Defaults to 5 for zip code conversions.

`remove_from_back` numeric, the number of digits to remove from the back of the string. If *NULL*, then removes characters from back more than specified in *length\_final*. Defaults to 4 for zip code conversions by removing the add-on codes.

### Value

vector, with characters formatted accordingly.

---

```
pretty_text          Removes spaces, special characters and capitals from string vector.
```

---

### Description

Removes paces, special characters and capitals from string vector and converts unknowns to NA.

### Usage

```
pretty_text(
  v,
  remove_after = FALSE,
  remove_punc = FALSE,
  remove_white = FALSE,
  add_na = TRUE
)
```

**Arguments**

v	vector, integer or sting vector with numerical values.
remove_after	boolean whether to remove text after -. Defaults to <i>FALSE</i> .
remove_punc	boolean, whether to remove punctuation marks. Defaults to <i>FALSE</i> .
remove_white	boolean, whether to remove white spaces. Defaults to <i>FALSE</i> .
add_na	boolean, whether to change text indicating NA to NA values in R. Defaults to <i>TRUE</i> .

**Value**

vector, with characters formatted accordingly.

---

remove_column	<i>Delete columns with all NA or all identical data.</i>
---------------	--

---

**Description**

Delete columns where all data elements are NA or the same.

**Usage**

```
remove_column(dt, na = TRUE, identical = TRUE)
```

**Arguments**

dt	data.table, to manipulate.
na	boolean, to delete columns where all data elements are NA.
identical	boolean, to delete columns where all data elements are the same.

**Value**

data table, with data.

# Index

[all\\_ids\\_mi2b2](#), 3

[convert\\_dia](#), 3  
[convert\\_enc](#), 5  
[convert\\_lab](#), 6  
[convert\\_med](#), 7  
[convert\\_notes](#), 9  
[convert\\_phy](#), 11  
[convert\\_prc](#), 12  
[convert\\_rfv](#), 13  
[create\\_img\\_db](#), 15

[export\\_notes](#), 16

[find\\_exam](#), 17

[load\\_all](#), 20  
[load\\_all\\_data](#), 22  
[load\\_bib](#), 23  
[load\\_con](#), 25  
[load\\_dem](#), 28  
[load\\_dem\\_old](#), 30  
[load\\_dia](#), 32  
[load\\_enc](#), 34  
[load\\_lab](#), 37  
[load\\_lno](#), 39  
[load\\_mcm](#), 41  
[load\\_med](#), 43  
[load\\_mic](#), 45  
[load\\_mrn](#), 47  
[load\\_notes](#), 49  
[load\\_phy](#), 51  
[load\\_prc](#), 53  
[load\\_prv](#), 55  
[load\\_ptd](#), 57  
[load\\_rdt](#), 59  
[load\\_rfv](#), 61  
[load\\_trn](#), 63

[pretty\\_mrn](#), 65

[pretty\\_numbers](#), 66

[pretty\\_text](#), 66

[remove\\_column](#), 67