Package 'dicomtk'

November 13, 2026

Type Package

Title Comprehensive 'DICOM' Toolkit for Medical Imaging

Version 0.1.0

Description A comprehensive R interface to the 'DICOM' Toolkit ('DCMTK'), providing native R bindings for reading, writing, modifying, and visualizing Digital Imaging and Communications in Medicine ('DICOM') files. Features include 'tidyverse'-compliant data structures, 'ggplot2'-based image visualization, interactive 'Shiny' viewers, 'PACS' networking (C-ECHO, C-FIND, C-STORE, C-MOVE), structured report handling, 'DICOM' deidentification, and integration with local and cloud-based large language models ('LLM') via 'Ollama' and 'OpenAI'-compatible APIs for AI-assisted medical image analysis. The 'DCMTK' C++ library is bundled and compiled from source, requiring no external installation.

License GPL (>= 2)

URL https://github.com/lauyeehow1986-hub/dicomtk, https://lauyeehow1986-hub.github.io/dicomtk/

BugReports https://github.com/lauyeehow1986-hub/dicomtk/issues

Depends R (>= 4.1.0)

Imports Rcpp (>= 1.0.12), tibble (>= 3.0.0), dplyr (>= 1.1.0), tidyr (>= 1.3.0), rlang (>= 1.1.0), ggplot2 (>= 3.4.0), cli (>= 3.6.0), httr2 (>= 1.0.0), jsonlite (>= 1.8.0), grDevices, graphics, stats, utils, tools

Suggests shiny (>= 1.8.0), rgl (>= 1.0.0), RNifti (>= 1.5.0), testthat (>= 3.0.0), knitr, rmarkdown, pkgdown, withr, magick

LinkingTo Rcpp

SystemRequirements C++17, CMake (>= 3.10)

Encoding UTF-8

RoxygenNote 7.3.3

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation yes

Author Yee How Lau [aut, cre],
John Muschelli [ctb] (Original dcmtk R wrapper),
OFFIS e.V. [cph] (DCMTK C++ library)

Maintainer Yee How Lau <lauyeehow1986@gmail.com>

Repository CRAN

Date/Publication 2026-11-13 18:00:14 UTC

Contents

create_dicom_sr . . . 3
datetime_tags . . . 4
date_tags . . . 4
dcm2json . . . 5
dcm2pdf . . . 5
dcm2xml . . . 6
dicom_anonymize . . . 6
dicom_ask . . . 8
dicom_compress . . . 8
dicom_connect . . . 9
dicom_decompress . . . 10
dicom_describe_image . . . 11
dicom_echo . . . 12
dicom_find . . . 12
dicom_get . . . 13
dicom_info . . . 14
dicom_llm . . . 14
dicom_modify . . . 16
dicom_qrdb_index . . . 17
dicom_receive . . . 17
dicom_report . . . 18
dicom_send . . . 19
dicom_summarize . . . 19
dicom_sync_upstream . . . 20
dicom_tags . . . 21
dicom_tag_lookup . . . 21
dicom_tag_value . . . 22
dicom_tls_config . . . 23
dicom_to_nifti . . . 24
dicom_to_png . . . 24
dicom_verify_signature . . . 25
dicom_viewer . . . 26
dicom_window . . . 26
dicom_worklist . . . 27
ggdicom . . . 28
have_dcmtk . . . 28

is_dicom 29
legacy-dcmtk 29
ollama_list_models 31
plot_dicom 31
read_dicom 32
read_dicom_ect 33
read_dicom_header 34
read_dicom_pixels 35
read_dicom_pmap 35
read_dicom_pstate 36
read_dicom_rt 37
read_dicom_seg 37
read_dicom_series 38
read_dicom_sr 39
read_dicom_tract 39
removal_tags 40
render_volume 40
write_dicom 41

Index **43**

`create_dicom_sr` *Create a DICOM Basic Text Structured Report (SR)*

Description

Writes a valid DICOM Basic Text SR Storage (`1.2.840.10008.5.1.4.1.1.88.11`) Part 10 file containing a structured report title, text content, and patient /study identifiers.

Usage

```
create_dicom_sr(
  out_path,
  title = "Radiology Structured Report",
  text,
  patient_name = "ANONYMOUS",
  patient_id = "ANON001",
  study_uid = ""
)
```

Arguments

`out_path`	Output `.dcm` file path.
`title`	Document / Series title (`SeriesDescription (0008,103E)`).
`text`	Report body text (`TextValue (0040,A160)`).
`patient_name`	Patient name (`(0010,0010)`). Default is `"ANONYMOUS"`.
`patient_id`	Patient ID (`(0010,0020)`). Default is `"ANON001"`.

study_uid Optional StudyInstanceUID to link the report to an existing study. If "", generates a new UID.

Value

Invisibly returns out_path.

Examples

```
sr_file <- tempfile(fileext = ".dcm")
create_dicom_sr(
  sr_file,
  title = "AI Radiology Summary",
  text = "FINDINGS: Unremarkable brain parenchyma. IMPRESSION: Normal head CT."
)
read_dicom_sr(sr_file)
```

---

datetime_tags *DICOM DateTime Tags*

Description

Standard DICOM tags with Value Representation DT (DateTime).

Usage

```
data(datetime_tags)
```

Format

A data.frame with columns tag and name.

---

date_tags *DICOM Date Tags*

Description

Standard DICOM tags with Value Representation DA (Date).

Usage

```
data(date_tags)
```

Format

A data.frame with columns tag and name.

dcm2json — *Convert a DICOM File to JSON (DICOM JSON Model PS3.18)*

Description

Exports the DICOM header and dataset elements to standard DICOMWeb JSON format (PS3.18).

Usage

```
dcm2json(file, out_path = NULL)
```

Arguments

file	Path to input DICOM file.
out_path	Optional output .json file path. If NULL (default), returns the JSON string directly.

Value

Character string of DICOM JSON (or invisibly out_path if written to a file).

Examples

```
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
json_txt <- dcm2json(dcm_file)
substr(json_txt, 1, 200)
```

dcm2pdf — *Extract Encapsulated PDF from a DICOM File*

Description

Extracts the binary PDF document stored in tag (0042,0011) EncapsulatedDocument.

Usage

```
dcm2pdf(file, out_path = tempfile(fileext = ".pdf"), opts = "", ...)
```

Arguments

file	Path to an Encapsulated PDF DICOM file.
out_path	Output .pdf path. Defaults to a temporary .pdf file.
opts	Optional arguments (kept for legacy compatibility).
...	Additional unused arguments.

Value

Character string of the output .pdf file path.

dcm2xml — *Convert a DICOM File to XML (Native DICOM Model PS3.19)*

Description

Exports the DICOM header and dataset elements to XML format.

Usage

```
dcm2xml(file, out_path = NULL, opts = "", ...)
```

Arguments

file	Path to input DICOM file.
out_path	Optional output .xml file path. If NULL (default), writes to a temporary .xml file and returns its path (with the XML string attached as attribute "xml").
opts	Optional arguments (ignored in native C++ mode, kept for legacy compatibility).
...	Additional unused arguments.

Value

Character string of the output .xml file path.

Examples

```
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
xml_file <- dcm2xml(dcm_file)
head(readLines(xml_file), 10)
```

dicom_anonymize — *De-identify / Anonymize a DICOM File (HIPAA & DICOM PS3.15 Profile)*

Description

Removes or replaces Protected Health Information (PHI) tags according to HIPAA Safe Harbor and the DICOM PS3.15 Basic Application Level Confidentiality Profile. Supports date shifting, private tag removal, UID regeneration, and custom tag overrides.

Usage

```
dicom_anonymize(
  path,
  out_path = NULL,
  patient_name = "ANONYMOUS",
  patient_id = "ANON001",
  shift_days = NULL,
  erase_private = TRUE,
  regenerate_uids = TRUE,
  extra_remove_tags = character(),
  replace_tags = list()
)
```

Arguments

path	Path to input DICOM file.
out_path	Output path for anonymized DICOM file. If NULL (default), writes to a temporary .dcm file.
patient_name	Replacement string for PatientName (0010,0010). Default is "ANONYMOUS".
patient_id	Replacement string for PatientID (0010,0020). Default is "ANON001".
shift_days	Optional integer number of days to shift all date (DA) tags (e.g., -365 shifts dates back by 1 year). If NULL (default), date tags are replaced with "19000101".
erase_private	Logical. Remove all odd-group private tags. Default is TRUE.
regenerate_uids	Logical. Regenerate SOPInstanceUID, StudyInstanceUID, and SeriesInstanceUID. Default is TRUE.
extra_remove_tags	Optional character vector of additional tags or keywords to remove.
replace_tags	Named list of additional tag/value replacements.

Value

Invisibly returns the path to the anonymized DICOM file.

Examples

```
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
anon_file <- dicom_anonymize(dcm_file, patient_name = "STUDY^SUBJECT01")
dicom_tag_value(anon_file, c("PatientName", "PatientIdentityRemoved"))
```

dicom_ask *Ask a Natural-Language Question About a DICOM File*

Description

Answers arbitrary questions about a DICOM file's metadata or image content using the configured LLM (or local metadata inspection).

Usage

```
dicom_ask(path, question, llm = dicom_llm("ollama"), include_image = FALSE)
```

Arguments

path	Path to a DICOM file.
question	Character string question (e.g., "What is the pixel spacing and modality?").
llm	A "dicom_llm" object created by dicom_llm().
include_image	Logical. Attach the rendered DICOM image for multimodal vision models. Default is FALSE.

Value

Character string answer.

Examples

```
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
dicom_ask(dcm_file, "What is the modality and image dimensions?", llm = dicom_llm("offline"))
```

dicom_compress *Compress a DICOM File (Lossless RLE)*

Description

Compresses native uncompressed pixel data into DICOM Lossless RLE transfer syntax (1.2.840.10008.1.2.5, per DICOM PS3.5 Annex G).

Usage

```
dicom_compress(path, out_path = tempfile(fileext = ".dcm"), method = c("rle"))
```

Arguments

path	Input DICOM file path.
out_path	Output compressed DICOM file path. Defaults to a temporary .dcm file.
method	Compression method ("rle").

Value

Invisibly returns `out_path`.

Examples

```
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
rle_file <- dicom_compress(dcm_file)
file.exists(rle_file)
```

`dicom_connect` *Create a DICOM PACS Network Connection Specification*

Description

Creates a `"dicom_connection"` object holding PACS / DICOM node connection parameters for use with `dicom_echo()`, `dicom_find()`, `dicom_get()`, and `dicom_send()`. Supports both DIMSE TCP connections and DICOMweb (QIDO-RS / WADO-RS / STOW-RS) HTTP endpoints.

Usage

```
dicom_connect(
  host = "127.0.0.1",
  port = 104L,
  called_ae = "ANY-SCP",
  calling_ae = "DICOMTK",
  timeout_ms = 5000L,
  dicomweb_url = NULL,
  local_dir = NULL
)
```

Arguments

`host`	Hostname or IP address of the remote DICOM peer (e.g., `"127.0.0.1"` or `"https://pacs.example.org/dicom-web"`).
`port`	Integer TCP port for DIMSE communication. Default is `104L` (or `4242L` for Orthanc).
`called_ae`	Remote Application Entity Title (Called AET). Default is `"ANY-SCP"`.
`calling_ae`	Local Application Entity Title (Calling AET). Default is `"DICOMTK"`.
`timeout_ms`	Socket timeout in milliseconds. Default is `5000L`.
`dicomweb_url`	Optional base URL for DICOMweb (QIDO-RS / WADO-RS / STOW-RS) REST APIs.
`local_dir`	Optional local directory path to act as a local file-backed PACS repository for testing queries and retrievals offline.

Value

An S3 object of class "dicom_connection".

Examples

```
conn <- dicom_connect("127.0.0.1", port = 4242, called_ae = "ORTHANC")
conn
```

`dicom_decompress` *Decompress an Encapsulated DICOM File*

Description

Decompresses RLE-compressed (or encapsulated) DICOM files into standard Explicit VR Little Endian (1.2.840.10008.1.2.1) uncompressed DICOM format.

Usage

```
dicom_decompress(path, out_path = tempfile(fileext = ".dcm"))
```

Arguments

path	Input compressed DICOM file path.
out_path	Output uncompressed DICOM file path. Defaults to a temporary .dcm file.

Value

Invisibly returns out_path.

Examples

```
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
rle_file <- dicom_compress(dcm_file)
dec_file <- dicom_decompress(rle_file)
is_dicom(dec_file)
```

`dicom_describe_image` *Describe a DICOM Image Using a Vision-Capable LLM*

Description

Renders the DICOM slice with clinical windowing to a PNG, encodes it as base64, and sends it along with anonymized technical metadata and pixel intensity statistics to a vision-capable LLM (e.g., Ollama `llava`, OpenAI `gpt-4o`, or Anthropic `claude-sonnet-4-20250514`).

Usage

```
dicom_describe_image(
  path,
  llm = dicom_llm("ollama"),
  frame = 1L,
  window = "auto",
  prompt = NULL
)
```

Arguments

`path`	Path to a DICOM file.
`llm`	A `"dicom_llm"` object created by `dicom_llm()`.
`frame`	Integer frame/slice index (1-based). Default is `1L`.
`window`	Window preset (`"auto"`, `"brain"`, `"lung"`, `"bone"`, etc.). Default is `"auto"`.
`prompt`	Custom instruction prompt for the vision model.

Value

A character string with the image description and quantitative stats.

Examples

```
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
llm <- dicom_llm("offline")
dicom_describe_image(dcm_file, llm = llm)
```

dicom_echo *Verify DICOM PACS Connectivity (C-ECHO SCU)*

Description

Performs a DICOM Upper Layer `A-ASSOCIATE-RQ` handshake followed by a Verification SOP Class (`1.2.840.10008.1.1`) `C-ECHO-RQ` over TCP, or verifies a local directory-backed PACS connection.

Usage

```
dicom_echo(
  x,
  port = 104L,
  called_ae = "ANY-SCP",
  calling_ae = "DICOMTK",
  timeout_ms = 5000L
)
```

Arguments

`x`	A `"dicom_connection"` object from `dicom_connect()` or a hostname string.
`port`	Integer port (used if `x` is a hostname string). Default is `104L`.
`called_ae`	Called AE Title. Default is `"ANY-SCP"`.
`calling_ae`	Calling AE Title. Default is `"DICOMTK"`.
`timeout_ms`	Timeout in milliseconds. Default is `5000L`.

Value

A tibble::tibble with columns `host`, `port`, `called_ae`, `success`, `status`, `message`, and `elapsed_ms`.

Examples

```
conn <- dicom_connect(local_dir = system.file("extdata", package = "dicomtk"))
dicom_echo(conn)
```

dicom_find *Query DICOM Studies, Series, or Instances (C-FIND / QIDO-RS / Local PACS)*

Description

Queries a DICOM connection or local PACS directory for matching studies, series, or instances, returning a tidy tibble::tibble that can be piped into `dplyr::filter()` and `dicom_get()`.

Usage

```
dicom_find(conn, level = c("STUDY", "SERIES", "IMAGE"), ...)
```

Arguments

conn	A "dicom_connection" object from dicom_connect(), or a local directory path.
level	Query level: "STUDY", "SERIES", or "IMAGE". Default is "STUDY".
...	Optional tag filters as named arguments (e.g., Modality = "CT", PatientName = "DOE*").

Value

A tibble::tibble of matching DICOM records.

Examples

```
ext_dir <- system.file("extdata", package = "dicomtk")
conn <- dicom_connect(local_dir = ext_dir)
dicom_find(conn)
```

---

dicom_get *Retrieve DICOM Files from PACS Connection (C-GET / C-MOVE / Local PACS)*

Description

Copies or downloads matching DICOM files from a "dicom_connection" or query result tibble into output_dir.

Usage

```
dicom_get(x, output_dir = tempfile("dicom_get_"), ...)
```

Arguments

x	A "dicom_connection" object or a tibble::tibble returned by dicom_find().
output_dir	Destination directory for retrieved .dcm files. Defaults to a temporary directory.
...	Optional filter criteria if x is a "dicom_connection".

Value

Character vector of retrieved file paths in output_dir.

Examples

```
ext_dir <- system.file("extdata", package = "dicomtk")
conn <- dicom_connect(local_dir = ext_dir)
files <- conn |> dicom_find() |> dicom_get()
length(files)
```

`dicom_info` *Inspect Bundled DCMTK Configuration and Version Info*

Description

Returns a tidy tibble describing the bundled DCMTK version, data dictionary edition, number of standard tags, and active C++ backend.

Usage

```
dicom_info()
```

Value

A tibble::tibble with DCMTK configuration details.

Examples

```
dicom_info()
```

`dicom_llm` *Configure an LLM Provider for DICOM Analysis*

Description

Creates a `"dicom_llm"` configuration object connecting `dicomtk` to:

- **Local Ollama** (`provider = "ollama"`, default `http://localhost:11434`)
- **OpenAI / OpenAI-compatible APIs** (`provider = "openai"` or `"custom"`, supporting OpenAI, OpenRouter, LM Studio, vLLM, Azure, or local proxy subscriptions)
- **Anthropic Claude** (`provider = "anthropic"`, supporting native `/v1/messages` as well as OpenAI-compatible endpoints)
- **Offline rule-based engine** (`provider = "offline"`, for deterministic local metadata & pixel-statistics summaries without external API calls)

Usage

```
dicom_llm(
  provider = c("ollama", "openai", "anthropic", "custom", "offline"),
  model = NULL,
  base_url = NULL,
  api_key = NULL,
  temperature = 0.2,
  max_tokens = 1024L,
  anonymize = TRUE,
  offline_fallback = TRUE
)
```

Arguments

provider	Character provider type: "ollama", "openai", "anthropic", "custom", or "offline". Default is "ollama".
model	Model identifier (e.g., "llava:13b", "llama3.2-vision", "gpt-4o", "claude-sonnet-4-20250514").
base_url	Optional API base URL. Defaults to "http://localhost:11434" for Ollama, "https://api.openai.com/v1" for OpenAI, and "https://api.anthropic.com/v1" for Anthropic.
api_key	API key or bearer token. Automatically reads from OPENAI_API_KEY or ANTHROPIC_API_KEY environment variables if not supplied.
temperature	Sampling temperature. Default is 0.2.
max_tokens	Maximum response tokens. Default is 1024L.
anonymize	Logical. Automatically strip HIPAA PHI tags before sending metadata to non-local endpoints. Default is TRUE.
offline_fallback	Logical. If TRUE (default), falls back to a deterministic local clinical summary if the remote/local server is unreachable.

Value

An S3 object of class "dicom_llm".

Examples

```
# Local Ollama configuration
local_llm <- dicom_llm("ollama", model = "llava:13b")

# OpenAI-compatible configuration
openai_llm <- dicom_llm("openai", model = "gpt-4o")

# Anthropic Claude configuration
claude_llm <- dicom_llm("anthropic", model = "claude-sonnet-4-20250514")
```

dicom_modify *Modify, Insert, or Delete DICOM Header Tags*

Description

Updates, inserts, or deletes specified tags in a DICOM file using the native C++ engine and writes the modified dataset to `out_path`.

Usage

```
dicom_modify(
  path,
  tags = list(),
  out_path = path,
  delete_tags = character(),
  erase_private = FALSE,
  regenerate_uids = FALSE
)
```

Arguments

`path`	Input DICOM file path.
`tags`	Named list or named character vector of tags (by `"(GGGG,EEEE)"` or keyword such as `"PatientName"`) and their new string values.
`out_path`	Output DICOM file path. Defaults to modifying `path` in-place.
`delete_tags`	Character vector of tags or keywords to remove.
`erase_private`	Logical. Remove all odd-group private tags. Default is `FALSE`.
`regenerate_uids`	Logical. Generate fresh `SOPInstanceUID`, `StudyInstanceUID`, and `SeriesInstanceUID`. Default is `FALSE`.

Value

Invisibly returns `out_path`.

Examples

```
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
tmp <- tempfile(fileext = ".dcm")
dicom_modify(dcm_file, tags = list(PatientName = "ANON^PATIENT"), out_path = tmp)
dicom_tag_value(tmp, "PatientName")
```

dicom_qrdb_index — *Build or Query a Local DICOM Query/Retrieve Index Database (dcmqrdb)*

Description

Scans a directory of DICOM files and builds a tidy index table organized by Patient, Study, Series, and Instance UIDs (dcmqrdb model).

Usage

```
dicom_qrdb_index(directory, recursive = TRUE)
```

Arguments

directory	Path to a directory containing DICOM files.
recursive	Logical. Scan subdirectories recursively. Default is TRUE.

Value

A tibble::tibble with columns file, PatientID, PatientName, StudyInstanceUID, StudyDate, Modality, SeriesInstanceUID, SOPInstanceUID, Rows, and Columns.

Examples

```
ext_dir <- system.file("extdata", package = "dicomtk")
dicom_qrdb_index(ext_dir)
```

dicom_receive — *Configure a Local DICOM Storage Receiver Directory (C-STORE SCP)*

Description

Sets up and returns a local directory-backed "dicom_connection" ready to receive, index, and query DICOM files.

Usage

```
dicom_receive(
  output_dir = tempfile("dicom_scp_"),
  port = 11112L,
  aetitle = "DICOMTK-SCP"
)
```

Arguments

output_dir	Directory to store received DICOM files.
port	Port number for metadata registration. Default is 11112L.
aetitle	Application Entity Title for the receiver. Default is "DICOMTK-SCP".

Value

A "dicom_connection" object configured with local_dir = output_dir.

Examples

```
scp <- dicom_receive(tempfile("scp_"))
scp
```

---

dicom_report *Generate a Structured Radiology Report from a DICOM File*

Description

Combines DICOM technical metadata and vision analysis to produce a structured radiology report (EXAMINATION, TECHNIQUE, FINDINGS, IMPRESSION), and optionally saves it as a DICOM Structured Report (.dcm) via create_dicom_sr().

Usage

```
dicom_report(
  path,
  llm = dicom_llm("ollama"),
  frame = 1L,
  window = "auto",
  save_sr_path = NULL
)
```

Arguments

path	Path to a DICOM file.
llm	A "dicom_llm" object created by dicom_llm().
frame	Integer frame/slice index (1-based). Default is 1L.
window	Window preset. Default is "auto".
save_sr_path	Optional output .dcm path to save the generated report as a formal DICOM Structured Report (Basic Text SR).

Value

A tibble::tibble with columns section and content, and the full formatted report text in attribute "report_text".

Examples

```
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
rpt <- dicom_report(dcm_file, llm = dicom_llm("offline"))
rpt
```

---

dicom_send *Send DICOM Files to a PACS Peer (C-STORE SCU)*

Description

Transmits one or more DICOM files to a configured "dicom_connection" (or local PACS repository directory).

Usage

```
dicom_send(files, conn)
```

Arguments

files	Character vector of DICOM file paths to send.
conn	A "dicom_connection" object from dicom_connect().

Value

A tibble::tibble summarizing the transmission status for each file.

Examples

```
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
pacs_dir <- tempfile("pacs_store_")
dir.create(pacs_dir)
conn <- dicom_connect(local_dir = pacs_dir)
dicom_send(dcm_file, conn)
```

---

dicom_summarize *Summarize a DICOM Study Using an LLM*

Description

Generates a concise clinical and technical natural-language summary of a DICOM file's header metadata (modality, anatomy, acquisition parameters, image geometry). Automatically strips PHI tags when communicating with non-local LLM endpoints.

Usage

```
dicom_summarize(path, llm = dicom_llm("ollama"))
```

Arguments

path	Path to a DICOM file.
llm	A "dicom_llm" object created by dicom_llm(). Defaults to dicom_llm("ollama").

Value

A character string containing the study summary.

Examples

```
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
llm <- dicom_llm("offline")
dicom_summarize(dcm_file, llm = llm)
```

---

dicom_sync_upstream *Check or Synchronize with Upstream OFFIS DCMTK Repository*

Description

Checks the official upstream DCMTK GitHub mirror (https://github.com/DCMTK/dcmtk) for new commits, dictionary releases, or C++ module header updates, and optionally synchronizes a local dicomtk source checkout by running tools/bundle_dcmtk.R.

Usage

```
dicom_sync_upstream(pkg_dir = ".", apply = FALSE, repo = "DCMTK/dcmtk")
```

Arguments

pkg_dir	Optional path to a local dicomtk package source directory when apply = TRUE. Defaults to ".".
apply	Logical; if TRUE, clones the latest DCMTK/dcmtk master branch and executes tools/bundle_dcmtk.R inside pkg_dir. Default is FALSE (check-only mode).
repo	GitHub repository slug. Default is "DCMTK/dcmtk".

Value

A tibble::tibble summarizing the bundled DCMTK version/commit, the latest upstream commit SHA/date/message from GitHub, and whether the bundle is up to date.

Examples

```
dicom_sync_upstream(apply = FALSE)
```

dicom_tags *Retrieve DICOM Tags from a File or the Standard Dictionary*

Description

If `path` is provided, reads and returns the DICOM header tags present in `path` as a tidy tibble. If `path` is `NULL`, returns the complete built-in OFFIS DCMTK 3.7.1 DICOM data dictionary (5,354 tags).

Usage

```
dicom_tags(path = NULL, ...)
```

Arguments

`path`	Optional file or directory path to read DICOM tags from. If `NULL` (default), returns the standard DICOM data dictionary.
`...`	Additional arguments passed to `read_dicom()` when `path` is not `NULL`.

Value

A tibble::tibble of DICOM tags.

Examples

```
# View standard DICOM dictionary
dicom_tags()

# Read tags from a DICOM file
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
dicom_tags(dcm_file)
```

dicom_tag_lookup *Search or Query the DICOM Data Dictionary*

Description

Searches the 5,354-entry DCMTK 3.7.1 DICOM data dictionary by substring or tag `(GGGG,EEEE)`. When an `llm` object (created by `dicom_llm()`) is supplied, natural-language questions (e.g., `"What tag stores tube current?"`) are also answered using the configured LLM.

Usage

```
dicom_tag_lookup(query, llm = NULL)
```

Arguments

query	Search string, tag "(GGGG,EEEE)", keyword, or natural-language question.
llm	Optional "dicom_llm" configuration object from dicom_llm() for natural-language tag interpretation.

Value

A tibble::tibble of matching DICOM dictionary entries (with an optional "llm_explanation" attribute if llm is provided).

Examples

```
dicom_tag_lookup("WindowCenter")
dicom_tag_lookup("(0028,1050)")
dicom_tag_lookup("Patient")
```

---

dicom_tag_value *Quickly Extract Specific Tag Values from a DICOM File*

Description

Retrieves one or more tag values by tag code (e.g., "(0010,0010)") or standard keyword (e.g., "PatientName", "Modality", "Rows").

Usage

```
dicom_tag_value(path, tags)
```

Arguments

path	Path to a single DICOM file.
tags	Character vector of tag IDs ("(GGGG,EEEE)") or DICOM keywords.

Value

A named character vector of tag values (NA_character_ if absent).

Examples

```
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
dicom_tag_value(dcm_file, c("Modality", "Rows", "Columns", "(0008,0070)"))
```

dicom_tls_config *Configure DICOM TLS Transport Security Parameters (*dcmtls*)*

Description

Attaches TLS certificate, private key, and CA bundle paths to a "dicom_connection" object for secure DICOM-over-TLS communication (RFC 3240 / DICOM PS3.15).

Usage

```
dicom_tls_config(
  conn,
  cert_file = NULL,
  key_file = NULL,
  ca_file = NULL,
  profile = c("BCP195", "BCP195-ND", "AES")
)
```

Arguments

conn	A "dicom_connection" object created by dicom_connect().
cert_file	Optional path to client X.509 PEM certificate.
key_file	Optional path to client private key.
ca_file	Optional path to trusted CA certificate bundle.
profile	DICOM TLS security profile ("BCP195", "BCP195-ND", or "AES"). Default is "BCP195".

Value

Updated "dicom_connection" object with TLS settings enabled.

Examples

```
conn <- dicom_connect("pacs.hospital.org", port = 2762) |>
  dicom_tls_config(profile = "BCP195")
conn
```

dicom_to_nifti *Convert DICOM File(s) or Series to NIfTI-1 Format*

Description

Converts a single DICOM file, vector of DICOM files, directory of DICOM slices, or "dicom_series" object into a standard NIfTI-1 (.nii or .nii.gz) volumetric file. Uses RNifti when installed or a built-in native NIfTI-1 binary writer with zero external dependencies.

Usage

```
dicom_to_nifti(path, out_path = NULL)
```

Arguments

path	Path to a DICOM file, directory, vector of files, or a "dicom_series" object.
out_path	Output .nii (or .nii.gz) path. If NULL, creates a temporary .nii file.

Value

Invisibly returns out_path.

Examples

```
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
nii_file <- dicom_to_nifti(dcm_file)
file.exists(nii_file)
```

dicom_to_png *Export a DICOM Image Slice to PNG*

Description

Renders a DICOM slice with window/level transformation to a PNG image file.

Usage

```
dicom_to_png(
  path,
  out_path = tempfile(fileext = ".png"),
  frame = 1L,
  window = "auto",
  width = NULL,
  height = NULL
)
```

Arguments

path	Path to a DICOM file, or a "dicom_pixels" matrix/array.
out_path	Output PNG file path. Defaults to a temporary .png file.
frame	Integer frame/slice index (1-based). Default is 1L.
window	Window preset name (e.g., "auto", "brain", "lung", "bone", "abdomen") or numeric c(center, width). Default is "auto".
width, height	Output PNG dimensions in pixels. Defaults to the native DICOM image dimensions.

Value

Invisibly returns out_path.

Examples

```
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
png_file <- dicom_to_png(dcm_file)
file.exists(png_file)
```

---

dicom_verify_signature

*Inspect Digital Signatures in a DICOM File (*dcmsign*)*

Description

Checks whether a DICOM dataset contains digital signatures in DigitalSignaturesSequence (FFFA,FFFA) or MACParametersSequence (4FFE,0001) and returns a tidy summary tibble.

Usage

```
dicom_verify_signature(path)
```

Arguments

path	Path to a DICOM file.

Value

A tibble::tibble of signature elements (0 rows if unsigned).

Examples

```
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
dicom_verify_signature(dcm_file)
```

dicom_viewer *Launch Interactive Shiny DICOM Viewer*

Description

Launches a Shiny web application for interactive exploration of DICOM files and multi-slice series, featuring slice scrolling, CT window/level presets, custom Window Center & Width sliders, colormap toggling, and a searchable DICOM header tag inspector.

Usage

```
dicom_viewer(path = NULL, launch.browser = interactive(), ...)
```

Arguments

`path`	Optional file or directory path to preload in the viewer. Defaults to the bundled example DICOM file.
`launch.browser`	Logical. Launch in browser/viewer pane. Default is `interactive()`.
`...`	Additional arguments passed to `shiny::runApp()`.

Value

Invisibly returns the Shiny application object (or runs the app if interactive).

Examples

```
if (interactive()) {
  dicom_viewer()
}
```

dicom_window *Apply VOI LUT Window/Level Transformation to DICOM Pixels*

Description

Applies linear window/level (Window Center and Window Width) transformation to a DICOM pixel matrix or array, normalizing output values to `[0, 1]`. Supports standard clinical CT window presets (`"brain"`, `"lung"`, `"bone"`, `"abdomen"`, `"mediastinum"`, `"liver"`, `"stroke"`, `"subdural"`, `"angio"`), embedded DICOM `WindowCenter`/`WindowWidth` tags (`"auto"`), or custom numeric `center` and `width`.

Usage

```
dicom_window(x, center = NULL, width = NULL, preset = "auto", invert = NULL)
```

Arguments

x	A "dicom_pixels" array/matrix or a file path to a DICOM file.
center	Numeric window center (level). Overrides preset if both center and width are provided.
width	Numeric window width (must be > 0).
preset	Character preset name ("auto", "brain", "stroke", "subdural", "lung", "mediastinum", "abdomen", "liver", "bone", "angio") or numeric vector c(center, width). Default is "auto".
invert	Optional logical to invert grayscale intensities. By default, automatically inverts if PhotometricInterpretation == "MONOCHROME1".

Value

A numeric matrix/array of the same dimensions as x with values scaled to [0, 1].

Examples

```
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
px <- read_dicom_pixels(dcm_file)
win_auto <- dicom_window(px, preset = "auto")
range(win_auto)
```

---

dicom_worklist *Query or Create Modality Worklist Entries (*dcmwlm*)*

Description

Reads or constructs a tidy Modality Worklist (MWL) schedule table containing ScheduledProcedureStepSequence (0040,0100) attributes (Modality, ScheduledStationAETitle, ScheduledProcedureStepStartDate, PatientName, PatientID, AccessionNumber).

Usage

```
dicom_worklist(path = NULL)
```

Arguments

path	Optional path to a DICOM worklist .dcm/.wl file or directory. If NULL, returns an empty template worklist tibble.

Value

A tibble::tibble representing Modality Worklist scheduled items.

Examples

```
dicom_worklist()
```

ggdicom *Create a ggplot2 DICOM Image Object*

Description

Alias for `plot_dicom()` designed for composable `ggplot2` pipelines.

Usage

```
ggdicom(x, frame = 1L, window = "auto", ...)
```

Arguments

`x`	Path to a DICOM file, a `"dicom_pixels"` matrix/array, or a `"dicom_series"` object.
`frame`	Integer slice/frame index (1-based) when `x` is multi-frame or a `"dicom_series"`. Default is `1L`.
`window`	Window preset (`"auto"`, `"brain"`, `"lung"`, `"bone"`, `"abdomen"`, etc.) or `c(center, width)`. Default is `"auto"`.
`...`	Additional arguments passed to `plot_dicom()`.

Value

A ggplot2::ggplot object.

Examples

```
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
ggdicom(dcm_file, palette = "viridis")
```

have_dcmtk *Check DCMTK Backend Availability and Version*

Description

Returns `TRUE` if the bundled DCMTK C++ engine and data dictionary are initialized and ready.

Usage

```
have_dcmtk()
```

Value

A logical scalar (`TRUE`).

Examples

```
have_dcmtk()
```

is_dicom *Check Whether File(s) Are Valid DICOM Files*

Description

Inspects the file header for the standard 128-byte preamble + "DICM" magic bytes or valid initial DICOM group/element tags.

Usage

```
is_dicom(path)
```

Arguments

path	Character vector of file paths.

Value

A logical vector of the same length as path.

Examples

```
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
is_dicom(dcm_file)
```

legacy-dcmtk *Legacy Compatibility Functions from* dcmtk

Description

Backward-compatible wrappers for functions in the original dcmtk R package (dcmdump, dcmdump_full, dcmhd, wide_hdr, burned_annot, dcmodify, dcm_anon, dcm_subtract_date, dcmj2pnm, dcmdjpeg, dcmdrle, dcmcrle, check_dcmtk_cmd), now powered by the native dicomtk C++17 engine without requiring external CLI binaries.

Usage

```
dcmdump(file, ...)

dcmdump_full(file, ...)

dcmhd(file, ...)

wide_hdr(hdr, ...)

burned_annot(file, ...)
```

```
dcmodify(file, tags = list(), outfile = file, ...)

dcm_anon(file, outfile = NULL, ...)

dcm_subtract_date(file, outfile = NULL, shift_days = -365L, ...)

dcmj2pnm(file, outfile = tempfile(fileext = ".png"), opts = "--write-png", ...)

dcmdjpeg(file, outfile = tempfile(fileext = ".dcm"), ...)

dcmdrle(file, outfile = tempfile(fileext = ".dcm"), ...)

dcmcrle(file, outfile = tempfile(fileext = ".dcm"), ...)

check_dcmtk_cmd(cmd = "dcmdump")
```

Arguments

file	Path to DICOM file(s).
...	Additional arguments passed to modern dicomtk equivalents.
hdr	A header data frame or tibble returned by read_dicom_header().
tags	Named list of tags and values to modify.
outfile	Optional output file path.
shift_days	Integer number of days to shift dates (default -365L).
opts	Optional command-line style options (ignored in native C++ mode).
cmd	Command name.

Value

Return values depend on the specific legacy wrapper function called:

dcmdump Character vector formatted similarly to CLI dcmdump output with tag, VR, value, length, VM, and keyword.

dcmdump_full A tibble::tibble containing full header and nested sequence items as returned by read_dicom().

dcmhd A tibble::tibble containing the top-level DICOM header elements.

wide_hdr A wide tibble::tibble with one row per file and columns corresponding to DICOM element keywords.

burned_annot Character scalar value of the BurnedInAnnotation (0028,0301) tag.

dcmodify Character vector of modified DICOM file paths as returned by dicom_modify().

dcm_anon, dcm_subtract_date Character vector of anonymized DICOM file paths as returned by dicom_anonymize().

dcmj2pnm Character path to the rendered output PNG file as returned by dicom_to_png().

dcmdjpeg, dcmdrle Character path to the decompressed DICOM file as returned by dicom_decompress().

dcmcrle Character path to the compressed DICOM file as returned by dicom_compress().

check_dcmtk_cmd Logical TRUE indicating the native DCMTK engine is available.

---

ollama_list_models *List Available Local Ollama Models*

Description

Queries the local Ollama daemon (GET /api/tags) and returns a tidy tibble::tibble of installed models (name, size_gb, modified_at). Returns an empty tibble if Ollama is not running.

Usage

```
ollama_list_models(
  base_url = Sys.getenv("OLLAMA_HOST", unset = "http://localhost:11434")
)
```

Arguments

base_url	Ollama base URL. Defaults to "http://localhost:11434".

Value

A tibble::tibble with columns name, size_gb, and modified_at.

Examples

```
ollama_list_models()
```

---

plot_dicom *Plot a DICOM Image Using ggplot2*

Description

Renders a 2D slice of a DICOM image as a ggplot2::ggplot object with clinical window/level scaling, proper pixel aspect ratio, and optional radiological metadata overlay.

Usage

```
plot_dicom(
  x,
  frame = 1L,
  window = "auto",
  center = NULL,
  width = NULL,
  palette = c("gray", "viridis", "inferno", "magma", "plasma"),
  annotate = TRUE,
  title = NULL
)
```

Arguments

x	Path to a DICOM file, a "dicom_pixels" matrix/array, or a "dicom_series" object.
frame	Integer slice/frame index (1-based) when x is multi-frame or a "dicom_series". Default is 1L.
window	Window preset ("auto", "brain", "lung", "bone", "abdomen", etc.) or c(center, width). Default is "auto".
center, width	Optional explicit window center and width.
palette	Color palette: "gray" (default radiological grayscale), "viridis", "inferno", "magma", or "plasma".
annotate	Logical. Display clinical overlay text (modality, dimensions, window center/width). Default is TRUE.
title	Optional plot title.

Value

A ggplot2::ggplot object.

Examples

```
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
p <- plot_dicom(dcm_file)
p
```

---

read_dicom *Read DICOM Metadata into a Tidy Tibble*

Description

Reads one or more DICOM files (or directories of DICOM files) using the native C++17 / DCMTK parser and returns a tidy tibble::tibble.

Usage

```
read_dicom(path, wide = FALSE, include_sequences = FALSE, recursive = TRUE)
```

Arguments

path	Character vector of DICOM file paths or directory paths.
wide	Logical. If FALSE (default), returns long format with one row per DICOM attribute (file, tag, group, element, vr, length, vm, keyword, name, value). If TRUE, pivots to wide format with one row per file and one column per DICOM keyword.
include_sequences	Logical. If TRUE, recursively extracts nested sequence (SQ) items. Default is FALSE.

recursive — Logical. When path contains directories, search subdirectories recursively. Default is TRUE.

Value

A tibble::tibble of class c("dicom_tbl", "tbl_df", "tbl", "data.frame").

Examples

```
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
hdr <- read_dicom(dcm_file)
hdr

# Wide format (1 row per file)
read_dicom(dcm_file, wide = TRUE)
```

---

read_dicom_ect — *Read DICOM Tractography Results (*dcmtract*) and Enhanced Multi-Frame IODs (*dcmect / dcmfg*)*

Description

Extracts functional groups (SharedFunctionalGroupsSequence (5200,9229), PerFrameFunctionalGroupsSequence (5200,9230)) and tractography sequences (TrackSetSequence (0066,0101)) into a tidy tibble::tibble.

Usage

```
read_dicom_ect(path)
```

Arguments

path — Path to a DICOM Enhanced CT or Tractography file.

Value

A tibble::tibble of functional group and tractography elements.

Examples

```
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
read_dicom_ect(dcm_file)
```

`read_dicom_header` *Read DICOM Header Metadata*

Description

Reads the header elements of one or more DICOM files into a tidy tibble. Compatible with both `dicomtk` tidy pipelines and legacy `dcmtk::read_dicom_header()` arguments.

Usage

```
read_dicom_header(
  file,
  replace_names = FALSE,
  add_opts = NULL,
  verbose = FALSE,
  wide = FALSE,
  ...
)
```

Arguments

`file`	Character vector of DICOM file paths or directory paths.
`replace_names`	Logical. Kept for compatibility with legacy `dcmtk`; tag names are automatically resolved against the built-in DCMTK 3.7.1 dictionary.
`add_opts`	Character vector of options (ignored in native C++ mode, kept for backward compatibility).
`verbose`	Logical. Print progress information. Default is `FALSE`.
`wide`	Logical. Return wide format (1 row per file). Default is `FALSE`.
`...`	Additional arguments passed to `read_dicom()`.

Value

A tibble::tibble containing the parsed DICOM header elements.

Examples

```
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
read_dicom_header(dcm_file)
```

read_dicom_pixels *Extract Pixel Data from a DICOM File*

Description

Reads and decodes 8-bit, 16-bit, or 32-bit monochrome or RGB pixel data (including RLE-compressed frames) directly into an R matrix or array, applying Modality LUT (`RescaleSlope` and `RescaleIntercept`) when `rescale = TRUE`.

Usage

```
read_dicom_pixels(path, frame = 0L, rescale = TRUE)
```

Arguments

`path`	Path to a single DICOM file.
`frame`	Integer frame number (1-indexed). Use `0` (default) to read all frames.
`rescale`	Logical. Apply `RescaleSlope` and `RescaleIntercept` to convert stored pixel values to modality units (e.g., Hounsfield Units for CT). Default is `TRUE`.

Value

A numeric matrix `[rows, cols]` (for single-frame monochrome) or 3D/4D array `[rows, cols, frames/channels]` with class `"dicom_pixels"` and DICOM imaging metadata attached as attributes.

Examples

```
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
px <- read_dicom_pixels(dcm_file)
dim(px)
```

read_dicom_pmap *Read DICOM Parametric Map Objects (*`dcmpmap`*)*

Description

Reads a DICOM Parametric Map (`PMAP`) dataset and applies `RealWorldValueSlope` and `RealWorldValueIntercept` transformations.

Usage

```
read_dicom_pmap(path)
```

Arguments

`path`	Path to a DICOM Parametric Map file.

Value

A list of class "dicom_pmap" containing values (real-world mapped numeric array) and metadata (tibble of header tags).

Examples

```
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
pmap <- read_dicom_pmap(dcm_file)
dim(pmap$values)
```

---

read_dicom_pstate *Read DICOM Grayscale Softcopy Presentation State (*dcmpstat*)*

Description

Extracts display shutter, VOI LUT, graphic annotation (0x0070), and spatial transformation parameters from a DICOM Presentation State (PR) object.

Usage

```
read_dicom_pstate(path)
```

Arguments

path	Path to a DICOM Presentation State file.

Value

A list of class "dicom_pstate" containing annotations (group 0x0070 elements), voi_lut (window/LUT attributes), and header.

Examples

```
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
ps <- read_dicom_pstate(dcm_file)
ps
```

read_dicom_rt *Read Radiation Therapy (RT) DICOM Objects (dcmrt)*

Description

Reads DICOM Radiation Therapy objects (RTPLAN, RTDOSE, RTSTRUCT, RTIMAGE), extracting group 0x3002, 0x3004, 0x3006, 0x3008, 0x300A, and 0x300C elements and nested sequences into a tidy tibble::tibble, along with dose grids if pixel data is present (RTDOSE).

Usage

```
read_dicom_rt(path)
```

Arguments

path	Path to a DICOM RT file.

Value

A list of class "dicom_rt" containing modality, summary (tidy tibble of top-level RT attributes), rt_elements (full hierarchical RT sequence tibble), and dose_grid (3D dose array if Modality == "RTDOSE").

Examples

```
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
rt_info <- read_dicom_rt(dcm_file)
rt_info$modality
```

read_dicom_seg *Read DICOM Segmentation Objects (dcmseg)*

Description

Reads a DICOM Segmentation (SEG) dataset, extracting segment descriptions (SegmentSequence (0062,0002)) and the binary/fractional 3D segmentation mask array.

Usage

```
read_dicom_seg(path)
```

Arguments

path	Path to a DICOM Segmentation file.

Value

A list of class "dicom_seg" containing segments (tibble of segment metadata) and mask (pixel/voxel mask array).

Examples

```
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
seg <- read_dicom_seg(dcm_file)
dim(seg$mask)
```

---

read_dicom_series *Read a Multi-Slice DICOM Series into a 3D Volume*

Description

Scans a directory (or vector of DICOM file paths), reads slice headers and pixel data, orders slices spatially by InstanceNumber or ImagePositionPatient, and assembles a 3D numeric array [rows, cols, slices] alongside a wide metadata tibble.

Usage

```
read_dicom_series(path, recursive = TRUE, rescale = TRUE)
```

Arguments

path	Directory path or character vector of DICOM file paths.
recursive	Logical. Search subdirectories recursively. Default is TRUE.
rescale	Logical. Apply RescaleSlope and RescaleIntercept. Default is TRUE.

Value

An object of class "dicom_series" (a named list with elements volume ([rows, cols, slices] numeric array), metadata (wide tibble of slice headers), and dims (integer vector)).

Examples

```
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
series <- read_dicom_series(c(dcm_file, dcm_file))
dim(series$volume)
```

read_dicom_sr *Read a DICOM Structured Report (SR) or Advanced IOD*

Description

Parses a DICOM Structured Report (`dcmsr`) or hierarchical DICOM dataset (including RT Plan/Dose/Structure Set, Segmentation, Parametric Map, and Tractography objects) into a tidy tibble::tibble preserving sequence hierarchy (`depth` and `sequence_path`).

Usage

```
read_dicom_sr(path)
```

Arguments

`path`	Path to a DICOM Structured Report (`.dcm`) file.

Value

A tibble::tibble with columns `depth`, `sequence_path`, `tag`, `keyword`, `vr`, and `value`.

Examples

```
sr_file <- tempfile(fileext = ".dcm")
create_dicom_sr(
  sr_file,
  title = "Chest CT Report",
  text = "No acute cardiopulmonary abnormality."
)
read_dicom_sr(sr_file)
```

read_dicom_tract *Read DICOM Tractography Results (`dcmtract`)*

Description

Reads a DICOM Tractography Results (`1.2.840.10008.5.1.4.1.1.66.6`) dataset, extracting `TrackSetSequence` (`0066,0101`) and streamline metadata into a tidy tibble::tibble.

Usage

```
read_dicom_tract(path)
```

Arguments

`path`	Path to a DICOM Tractography file.

Value

A list of class "dicom_tract" containing track_sets (tibble of group 0x0066 elements) and header (top-level metadata tibble).

Examples

```
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
tr <- read_dicom_tract(dcm_file)
tr
```

---

removal_tags *HIPAA / DICOM De-identification Removal Tags*

Description

Standard DICOM tags containing Protected Health Information (PHI) targeted for removal or replacement during anonymization.

Usage

```
data(removal_tags)
```

Format

A data.frame with columns tag and name.

---

render_volume *Render a 3D DICOM Volume (Orthogonal MPR, MIP, or Interactive rgl 3D)*

Description

Visualizes a 3D DICOM series ("dicom_series" or 3D numeric array) using:

- "mpr": 3-plane Orthogonal Multi-Planar Reconstruction (Axial, Coronal, Sagittal slices) via ggplot2.
- "mip": Maximum Intensity Projection across Axial, Coronal, and Sagittal orientations via ggplot2.
- "rgl": Interactive 3D voxel rendering using the rgl package (falls back to "mip" if rgl is not installed).

Usage

```
render_volume(
  x,
  method = c("mpr", "mip", "rgl"),
  window = "auto",
  threshold = 0.5,
  palette = c("gray", "viridis", "inferno", "magma", "plasma")
)
```

Arguments

x	A "dicom_series" object (from read_dicom_series()), a 3D numeric array [rows, cols, slices], or a directory/vector of DICOM file paths.
method	Visualization mode: "mpr" (default), "mip", or "rgl".
window	Window preset ("auto", "brain", "lung", "bone", etc.) or c(center, width). Default is "auto".
threshold	Optional intensity threshold in [0, 1] for 3D "rgl" point-cloud / surface rendering. Default is 0.5.
palette	Color palette ("gray", "viridis", "inferno", "magma", "plasma"). Default is "gray".

Value

A ggplot2::ggplot object (for "mpr" and "mip") or invisibly the rgl scene ID (for "rgl").

Examples

```
dcm_file <- system.file("extdata", "example.dcm", package = "dicomtk")
series <- read_dicom_series(c(dcm_file, dcm_file, dcm_file))
render_volume(series, method = "mpr")
render_volume(series, method = "mip")
```

---

write_dicom *Write a Matrix, Array, or DICOM Series to a DICOM File*

Description

Creates a standard DICOM Part 10 Explicit VR Little Endian file from an R 2D matrix or 3D array of pixel values and an optional named list or character vector of DICOM metadata tags.

Usage

```
write_dicom(x, path, metadata = list())
```

Arguments

x	A numeric 2D matrix [rows, cols], 3D array [rows, cols, frames], or "dicom_pixels" object.
path	Output file path for the written .dcm file.
metadata	Named list or named character vector of DICOM tags/keywords and values (e.g., list(PatientName = "DOE^JOHN", Modality = "CT")).

Value

Invisibly returns the normalized output path.

Examples

```
mat <- matrix(seq(0, 255), nrow = 16, ncol = 16)
out <- tempfile(fileext = ".dcm")
write_dicom(mat, out, metadata = list(PatientName = "TEST^PATIENT", Modality = "CT"))
read_dicom(out, wide = TRUE)
```

Index

∗ **datasets**
  date_tags, 4
  datetime_tags, 4
  removal_tags, 40

burned_annot *(*legacy-dcmtk*)*, 29

check_dcmtk_cmd *(*legacy-dcmtk*)*, 29
create_dicom_sr, 3
create_dicom_sr(), *18*

date_tags, 4
datetime_tags, 4
dcm2json, 5
dcm2pdf, 5
dcm2xml, 6
dcm_anon *(*legacy-dcmtk*)*, 29
dcm_subtract_date *(*legacy-dcmtk*)*, 29
dcmcrle *(*legacy-dcmtk*)*, 29
dcmdjpeg *(*legacy-dcmtk*)*, 29
dcmdrle *(*legacy-dcmtk*)*, 29
dcmdump *(*legacy-dcmtk*)*, 29
dcmdump_full *(*legacy-dcmtk*)*, 29
dcmhd *(*legacy-dcmtk*)*, 29
dcmj2pnm *(*legacy-dcmtk*)*, 29
dcmodify *(*legacy-dcmtk*)*, 29
dicom_anonymize, 6
dicom_anonymize(), *30*
dicom_ask, 8
dicom_compress, 8
dicom_compress(), *31*
dicom_connect, 9
dicom_connect(), *12, 13*, *19*, *23*
dicom_decompress, 10
dicom_decompress(), *30*
dicom_describe_image, 11
dicom_echo, 12
dicom_echo(), *9*
dicom_find, 12
dicom_find(), *9*, *13*
dicom_get, 13
dicom_get(), *9*, *12*
dicom_info, 14
dicom_llm, 14
dicom_llm(), *8*, *11*, *18*, *20–22*
dicom_modify, 16
dicom_modify(), *30*
dicom_qrdb_index, 17
dicom_receive, 17
dicom_report, 18
dicom_send, 19
dicom_send(), *9*
dicom_summarize, 19
dicom_sync_upstream, 20
dicom_tag_lookup, 21
dicom_tag_value, 22
dicom_tags, 21
dicom_tls_config, 23
dicom_to_nifti, 24
dicom_to_png, 24
dicom_to_png(), *30*
dicom_verify_signature, 25
dicom_viewer, 26
dicom_window, 26
dicom_worklist, 27

ggdicom, 28
ggplot2::ggplot, *28*, *31*, *32*, *41*

have_dcmtk, 28

is_dicom, 29

legacy-dcmtk, 29

ollama_list_models, 31

plot_dicom, 31
plot_dicom(), *28*

read_dicom, 32

read_dicom(), *21*, *30*, *34*
read_dicom_ect, 33
read_dicom_header, 34
read_dicom_header(), *30*
read_dicom_pixels, 35
read_dicom_pmap, 35
read_dicom_pstate, 36
read_dicom_rt, 37
read_dicom_seg, 37
read_dicom_series, 38
read_dicom_series(), *41*
read_dicom_sr, 39
read_dicom_tract, 39
removal_tags, 40
render_volume, 40

shiny::runApp(), *26*

tibble::tibble, *12–14*, *17–22*, *25*, *27*, *30–34*, *37*, *39*

wide_hdr *(*legacy-dcmtk*)*, 29
write_dicom, 41