

Package: cdisdata (via r-universe)

October 24, 2026

Title Versioned CDISC Standards Reference Data

Version 0.2.0

Description Provides versioned reference data for the standards of the Clinical Data Interchange Standards Consortium (CDISC) <https://www.cdisc.org/standards>: the CDISC Controlled Terminology (CT) for the Study Data Tabulation Model (SDTM) and the Analysis Data Model (ADaM); variable metadata for the SDTM model and for the families of the SDTM Implementation Guide (SDTMIG), the Standard for Exchange of Nonclinical Data Implementation Guide (SENDIG), and the ADaM Implementation Guide (ADaMIG); the Clinical Data Acquisition Standards Harmonization (CDASH) model and implementation guide; the Questionnaires, Ratings, and Scales (QRS) supplements; and the Define-XML schemas, written in the Extensible Markup Language (XML) Schema Definition (XSD) language, with Extensible Stylesheet Language Transformations (XSLT) stylesheets for them. A validity-date design stores every historical CT version compactly, without duplicating rows. Intended as a shared data dependency for pharmaverse-aligned packages <https://pharmaverse.org>.

License MIT + file LICENSE

Depends R (>= 4.1.0)

Suggests testthat (>= 3.0.0), xml2, withr, covr, pkgdown, knitr, rmarkdown, callr

LazyData true

LazyDataCompression xz

Encoding UTF-8

Language en-US

Config/testthat/edition 3

URL <https://github.com/humanpred/cdisdata>,
<https://humanpred.github.io/cdisdata/>

BugReports <https://github.com/humanpred/cdisdata/issues>

VignetteBuilder knitr

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Bill Denney [aut, cre] (ORCID:
<https://orcid.org/0000-0002-5759-428X>), Clinical Data
 Interchange Standards Consortium (CDISC) [cph] (Define-XML
 schema files in inst/extdata/schema (MIT License)), Lex Jansen
 [cph] (Define-XML stylesheets in inst/extdata/stylesheet (MIT
 License)), National Cancer Institute Enterprise Vocabulary
 Services [dct] (CDISC Controlled Terminology (public domain))

Maintainer Bill Denney <wdenney@humanpredictions.com>

Repository <https://cran.r-universe.dev>

Date/Publication 2026-10-24 16:40:09 UTC

RemoteUrl <https://github.com/cran/cdiscdata>

RemoteRef HEAD

RemoteSha 4a27ab2699bad1e7ba9e9e970d6f45ed6299c179

Contents

available_ct_versions	3
build_domain_spec	3
cdash_model	5
cdiscdata_versions	6
ct_adam	7
ct_sdtm	7
datasets_catalogue	8
get_cdash	9
get_ct	10
get_dataset	10
get_ig	11
ig_adam	12
ig_cdash	13
ig_sdtm	15
ig_sources	16
list_datasets	17
model_sdtm	17
qrs_supplement	18
schema_path	19
stylesheet_path	19

Index	21
--------------	-----------

available_ct_versions *List available CT versions*

Description

List available CT versions

Usage

```
available_ct_versions(type = c("sdm", "adam"))
```

Arguments

type One of "sdm" or "adam".

Value

Character vector of version date strings, newest first, e.g. c("2026-03-27", "2025-09-27", ...).

Examples

```
available_ct_versions("sdm")
available_ct_versions("adam")
```

build_domain_spec *Build a variable specification for PP, SUPPPP, or ADPP*

Description

Combines [get_ig](#) (variable name, label, type, core, order, codelist) with [get_ct](#) (is that codelist in the controlled terminology release) into a single, ready-to-use variable specification for one of the three PK datasets. PP and SUPPPP draw on an SDTM-side implementation guide (SUPPPP from the generic SUPP- structure, since CDISC has no domain-specific SUPPPP table); ADPP draws on the ADaMIG Basic Data Structure (BDS) table, since ADPP is a BDS dataset and CDISC has no ADPP-specific table either. An ADPP spec can optionally be unioned with the ADaMIG ADSL table (adsl), with a BDS extension such as ADaMIG-NCA (extension), and with the SDTM PP domain's variables (sdm_domain).

Usage

```
build_domain_spec(
  domain = c("PP", "SUPPPP", "ADPP"),
  ig_version = NULL,
  ct_version = NULL,
  adsl = TRUE,
  sdm_domain = NULL,
```

```

    sdtmig_version = NULL,
    standard = NULL,
    extension = NULL,
    extension_version = NULL
)

```

Arguments

domain	One of "PP", "SUPPPP", or "ADPP".
ig_version	Version of standard to use. NULL uses the newest available.
ct_version	CT version to check codelist ids against; passed to get_ct . NULL uses the newest available.
adsl	For domain = "ADPP" only: whether to union in the ADaMIG ADSL variables (default TRUE), since a real ADPP dataset carries ADSL's subject-level variables (treatment, demographics, ...) alongside its own BDS variables. FALSE returns the BDS variables alone. Ignored (with a warning) for domain != "ADPP".
sdtm_domain	For domain = "ADPP" only: NULL (the default) adds nothing; "PP" also unions in the SDTMIG PP domain's variables, the same way adsl = TRUE unions ADSL's: marked source = "SDTMIG", core forced to "Perm", a variable already defined keeping that version, and codelist ids checked against the ADaM CT first and the SDTM CT second. Ignored (with a warning) for domain != "ADPP"; any value other than "PP" is an error.
sdtmig_version	SDTMIG version to take the sdtm_domain variables from. NULL uses the newest available. Ignored (with a warning) unless domain = "ADPP" and sdtm_domain is set.
standard	The implementation guide to take the domain's variables from. NULL (the default) uses "SDTMIG" for PP and SUPPPP and "ADaMIG" for ADPP; any standard in get_ig that has the domain works (e.g. "SENDIG" defines PP too).
extension	For domain = "ADPP" only: NULL (the default) adds nothing; "ADaMIG-NCA" (or the alias "NCA"), "ADaM-popPK", or "ADaM-BDS-TTE" unions that BDS extension onto the BDS variables. Unlike the ADSL and PP unions, the extension's own Core designations are kept, as they are the point of the extension: a new variable is appended with its published Core, and a variable the BDS already defines takes the extension's (stronger) Core (e.g. AVISIT becomes required in ADaMIG-NCA). Rows the extension touches are marked with its name in source. Ignored (with a warning) for domain != "ADPP".
extension_version	Version of the extension; NULL uses the newest. Ignored (with a warning) unless domain = "ADPP" and extension is set.

Details

Neither the ADaMIG BDS nor ADSL tables define the PP-inherited traceability variables (PPTTESTCD, PPTTEST, and the rest of PP's variables that a BDS dataset built from PP typically carries forward); which of PP's variables to carry into ADPP, and under what names, is a downstream derivation choice (e.g. admiral's own conventions), so they are only added when asked for with `sdtm_domain`, and the default leaves them out.

Value

A data frame with columns `variable`, `label`, `type`, `length`, `core`, `order`, `source`, and `codelist_id`. `source` is "SDTMIG" (or the SDTM-side standard) for PP/SUPPPP; "BDS", "ADSL", "SDTMIG", or the extension's name for ADPP. `order` is the published order, renumbered consecutively, with unioned variables following. `codelist_id` is the C-code of the first codelist the IG lists for the variable, when that codelist is in the CT release used, and NA otherwise (no codelist, or one not in that release, e.g. a codelist since retired). `length` is always NA: the CDISC Library exports this package is built from carry no length, which is a sponsor choice, and the column is kept so existing code that reads it keeps working. ADSL-sourced ADPP rows have `core` forced to "Perm" regardless of their Core designation in ADSL itself: from ADPP's perspective, merging in an ADSL variable is a common but optional choice, not a requirement ADSL's own Core reflects.

Examples

```
build_domain_spec("PP")
build_domain_spec("SUPPPP")
build_domain_spec("ADPP") # BDS + ADSL (default)
build_domain_spec("ADPP", adsl = FALSE) # BDS only
build_domain_spec("ADPP", sdtm_domain = "PP") # BDS + ADSL + PP variables
build_domain_spec("ADPP", extension = "NCA") # BDS + ADSL + ADaMIG-NCA
```

cdash_model

*CDASH model variable metadata***Description**

One row per variable of the CDISC Clinical Data Acquisition Standards Harmonization (CDASH) model, at every version: 1.0, 1.1, 1.2, and 1.3. Built from CDISC Library CSV exports not redistributed with this package (see [ig_sources](#)); only variable metadata and the collection wording (question text, prompt) is carried, not the model's definitions, mapping instructions, or implementation notes. Retrieve it with `get_cdash("CDASH")`.

Usage

```
cdash_model
```

Format

A data frame with columns:

standard Always "CDASH".

version Model version, e.g. "1.3".

class Observation or special-purpose class, e.g. "Findings", "Identifiers".

domain The domain a domain-specific variable belongs to (e.g. "AE"), or NA for class-level variables.

order Position within the class and domain, as published.

variable CDASH variable name, e.g. "--TERM".

label Variable label. At most 40 characters except for 6 labels the model publishes longer (--TESTCD in 1.0 and 1.1; --ENDATF in all four versions).

domain_specific TRUE where the model flags the variable as domain specific; NA where it does not say.

question_text The CRF question text the model suggests.

prompt The CRF field prompt the model suggests.

type "Char" or "Num".

sdtm_target The SDTM variable or variables the collected value maps to.

codelist_code The CDISC CT codelist C-code the variable uses; NA when none.

Details

A (standard, version, class, domain, variable) is unique. The model defines most variables once per class with a -- prefix (domain is NA) and some per domain.

Source

CDISC Library CSV exports, downloaded under CDISC's terms and not redistributed; see `data-raw/README.md`.

cdiscdata_versions	<i>Summarise all bundled data versions</i>
--------------------	--

Description

Summarise all bundled data versions

Usage

```
cdiscdata_versions()
```

Value

A data frame with one row per data type showing the latest version, number of versions available, and when the data was last updated.

Examples

```
cdiscdata_versions()
```

ct_adam	<i>ADaM Controlled Terminology</i>
---------	------------------------------------

Description

Versioned CDISC ADaM Controlled Terminology from the NCI EVS FTP site. Same validity-date design as [ct_sdtm](#).

Usage

ct_adam

Format

A data frame with the same columns as [ct_sdtm](#).

Source

<https://evs.nci.nih.gov/ftp1/CDISC/ADaM/>

ct_sdtm	<i>SDTM Controlled Terminology</i>
---------	------------------------------------

Description

Versioned CDISC SDTM Controlled Terminology from the NCI EVS FTP site. Uses a validity-date design: one row per unique term-state across all versions. A term is "current" when valid_to is NA.

Usage

ct_sdtm

Format

A data frame with columns:

codelist_code NCI C-code for the codelist, e.g. "C66731".

codelist_name Submission value for the codelist, e.g. "SEX".

codelist_label Long name, e.g. "Sex".

extensible Logical; whether sponsor extensions are allowed.

term_code NCI C-code for the term, e.g. "C16576". NA for codelist-level rows.

term Submission value, e.g. "F". NA for codelist-level rows.

decoded_value NCI preferred term / decoded value, e.g. "Female".

synonyms Pipe-separated synonyms.

definition NCI definition text.

valid_from Date of the first CT release in which this row's content appeared.

valid_to Date of the last CT release in which this row's content was present; NA if still current.

Source

<https://evs.nci.nih.gov/ftp1/CDISC/SDTM/>

datasets_catalogue	<i>Datasets catalogue</i>
--------------------	---------------------------

Description

Metadata about all datasets bundled in cdisccdata, including CT tables and Define-XML file-based assets.

Usage

datasets_catalogue

Format

A data frame with columns:

dataset R object name or logical dataset identifier.

type One of "CT", "IG", "Model", "CDASH", "QRS", "Schema", "Stylesheet".

ct_type One of "sdm", "adam", or NA for non-CT datasets.

description Human-readable description.

versions Human-readable version range string.

n_versions Number of distinct versions available.

latest Most recent version string.

last_updated Date this catalogue row was last refreshed.

get_cdash

*Retrieve CDASH model or implementation-guide variable metadata***Description**

Accessor over the two CDASH (collection-side) tables, [cdash_model](#) and [ig_cdash](#), with the same conventions as [get_ig](#): the standard is chosen with `standard` and the version defaults to its newest. CDASH is kept apart from `get_ig()` because its tables describe case report form collection (question text, prompts, the SDTM variable each field maps to), not a dataset's structure, and so have different columns.

Usage

```
get_cdash(standard = "CDASHIG", version = NULL, domain = NULL)
```

Arguments

<code>standard</code>	"CDASHIG" (default) or "CDASH".
<code>version</code>	A version string of that standard (e.g. "2.3"). NULL (the default) uses the newest, compared numerically.
<code>domain</code>	Restrict to one domain (e.g. "LB"); for "CDASH" a class (e.g. "Findings") is accepted too. NULL (the default) applies no filter.

Details

The standards: "CDASH" (the CDASH model, versions 1.0 to 1.3, in [cdash_model](#)) and "CDASHIG" (the CDASH implementation guide, 1.1 and 2.0 to 2.3, in [ig_cdash](#)).

Like the other implementation-guide data these come from CDISC Library CSV exports that are not part of this package or repository (see [ig_sources](#)); the guides' definitions, CRF completion instructions, mapping instructions, and implementation notes are not carried. The collection wording the guides publish (`question_text`, `prompt`) is.

Value

A data frame: the rows of [cdash_model](#) or [ig_cdash](#) for that standard, version, and domain. An unknown standard, version, or domain is a classed error (`cdiscdata_error_ig_standard_unavailable`, `cdiscdata_error_ig_version_unavailable`, `cdiscdata_error_ig_domain_unavailable`).

Examples

```
get_cdash() # newest CDASHIG (2.3)
get_cdash("CDASHIG", version = "2.1", domain = "LB")
get_cdash("CDASH", domain = "Findings")
```

get_ct	Retrieve CDISC Controlled Terminology
--------	---------------------------------------

Description

Convenience wrapper around [get_dataset](#) for CT data.

Usage

```
get_ct(type = c("sdm", "adam"), version = NULL)
```

Arguments

type	One of "sdm" or "adam".
version	CT version date string e.g. "2024-09-27". NULL returns the latest available version.

Value

A data frame of CT terms valid at the requested version.

Examples

```
ct <- get_ct("sdm")
head(ct)
ct_2024 <- get_ct("sdm", version = "2024-09-27")
nrow(ct_2024)
head(get_ct("adam"))
```

get_dataset	Retrieve any cdisccdata dataset by name and version
-------------	---

Description

Generic retrieval gateway. For CT datasets, filters to rows valid at the requested version date. For the implementation-guide, model, CDASH, and QRS datasets (ig_sdm, model_sdm, ig_adam, cdash_model, ig_cdash, qrs_supplement), returns the whole table, or the rows whose version column equals version (across every standard or instrument in it; use [get_ig](#) or [get_cdash](#) to pick one standard). For schemas and stylesheets, returns the file path for the requested Define-XML version. Use [list_datasets](#) to see available dataset names.

Usage

```
get_dataset(name, version = NULL)
```

Arguments

name	Dataset name from <code>list_datasets()</code> \$dataset.
version	Version string. For CT, NULL returns the latest available version and a value is a date string e.g. "2024-09-27". For schemas/stylesheets, NULL is the latest and a value is a Define-XML version string e.g. "2.1". For the IG, model, CDASH, and QRS datasets, NULL returns every version.

Value

A data frame (for CT, IG, model, CDASH, and QRS data) or a file path string (for schemas/stylesheets).

Examples

```
ct <- get_dataset("ct_sdtm")
head(ct)
ct_2024 <- get_dataset("ct_sdtm", version = "2024-09-27")
nrow(ct_2024)
ig_adam_1_3 <- get_dataset("ig_adam", version = "1.3")
nrow(ig_adam_1_3)
get_dataset("define_xml_schema", version = "2.1")
get_dataset("define_xml_stylesheet", version = "2.0")
```

get_ig	<i>Retrieve CDISC implementation-guide or model variable metadata</i>
--------	---

Description

One accessor over every standard in `ig_sdtm`, `model_sdtm`, and `ig_adam`, mirroring `get_ct`. The standard is chosen with `standard`; the version defaults to that standard's newest.

Usage

```
get_ig(standard = "SDTMIG", version = NULL, domain = NULL)
```

Arguments

standard	One standard from the list above (default "SDTMIG"), or the alias "sdtm" or "adam".
version	A version string of that standard (e.g. "3.4"). NULL (the default) uses the newest, compared numerically, so "3.1.3" ranks below "3.2".
domain	Restrict to one domain or structure. For the implementation-guide standards in <code>ig_sdtm</code> , the domain column (e.g. "PP"); for "SDTM", a class (e.g. "Findings") or a dataset (e.g. "DM"); for the ADaM standards, the structure column, where "BDS" and "ADSL" are accepted as aliases for "Basic Data Structure" and "Subject-Level Analysis Dataset". NULL (the default) applies no filter.

Details

The standards, and the dataset each is stored in:

- "SDTM" (model versions 1.2 to 2.1): [model_sdtm](#).
- "SDTMIG" (3.1.2 to 3.4), "SDTMIG-AP", "SDTMIG-MD", "SENDIG" (3.0 to 3.1.1), "SENDIG-AR", "SENDIG-DART", "SENDIG-GeneTox": [ig_sdtm](#).
- "ADaMIG" (1.0 to 1.3), "ADaMIG-MD", "ADaMIG-NCA", "ADaM-ADAE", "ADaM-BDS-TTE", "ADaM-OCCDS", "ADaM-popPK": [ig_adam](#).

The lower-case names "sdtm" and "adam", the only values the first argument took before it named a standard, are kept as aliases for the defaults "SDTMIG" and "ADaMIG" (note "SDTM" in capitals is the model, not the implementation guide).

All of this comes from CDISC Library CSV exports that are not part of this package or repository (see [ig_sources](#) for exactly which files, and data-raw/README.md). Only variable metadata is carried; the guides' prose (CDISC Notes, Description, Definition, Examples) is not.

Value

A data frame: the rows of [ig_sdtm](#), [model_sdtm](#), or [ig_adam](#) for that standard, version, and domain.

An unknown standard, version, or domain is a classed error (`cdiscdata_error_ig_standard_unavailable`, `cdiscdata_error_ig_version_unavailable`, `cdiscdata_error_ig_domain_unavailable`).

Examples

```
get_ig() # newest SDTMIG (3.4)
get_ig("SDTMIG", version = "3.3", domain = "PP")
get_ig("SENDIG", domain = "PP")
get_ig("ADaMIG", domain = "BDS")
get_ig("ADaMIG-NCA")
get_ig("SDTM", version = "2.1", domain = "Findings")
```

ig_adam

ADaM implementation-guide variable metadata

Description

One row per variable of seven ADaM standards, at every version: ADaMIG (1.0, 1.1, 1.2, 1.3), ADaMIG-MD (1.0), ADaMIG-NCA (1.0), ADaM-ADAE (1.0), ADaM-BDS-TTE (1.0), ADaM-OCCDS (1.0, 1.1), and ADaM-popPK (1.0); 11 standard versions in all. Built from CDISC Library CSV exports that are not part of this package or repository (see [ig_sources](#) and data-raw/README.md); only variable metadata is carried, not the guides' prose. Use [get_ig](#) to retrieve a standard, and [build_domain_spec](#) to build an ADPP variable spec from it (ADPP is a Basic Data Structure dataset; ADaMIG-NCA extends it).

Usage

```
ig_adam
```

Format

A data frame with columns:

standard The standard, e.g. "ADaMIG" or "ADaMIG-NCA".

version The standard's version, e.g. "1.3".

structure The data structure, as named in the export, e.g. "Basic Data Structure" or "Subject-Level Analysis Dataset" (ADSL).

variable_set The variable set within the structure, e.g. "Timing".

order Row order within the export for that standard version.

variable Variable name, e.g. "AVAL".

label Variable label. At most 40 characters except PBCHGCyN in ADaMIG 1.2 and 1.3, which the guide publishes at 41.

type "Char" or "Num".

core Core designation ("Req", "Perm", "Cond", ...).

codelist_code CDISC CT codelist C-code(s) the variable uses; several are separated by "; ". NA when none.

codelist_submission_values The codelist submission value(s), where the export gives them.

described_value_domain A described value domain, where the variable has one rather than a codelist.

value_list A fixed list of allowed values, where there is one.

Details

A (standard, version, structure, variable_set, variable) is unique: the variable set is part of the key because ADaM-OCCDS defines DECDORGw twice, once for each dictionary-specific variable set, with different labels.

Source

CDISC Library CSV exports, downloaded under CDISC's terms and not redistributed; see `data-raw/README.md`.

 ig_cdash

CDASH implementation-guide variable metadata

Description

One row per variable of every domain and data collection scenario of the CDASH implementation guide (CDASHIG), at every version: 1.1, 2.0, 2.1, 2.2, and 2.3. Built from CDISC Library CSV exports not redistributed with this package (see [ig_sources](#)); only variable metadata and the collection wording (question text, prompt) is carried, not the guide's definitions, CRF completion instructions, mapping instructions, or implementation notes. Retrieve it with [get_cdash](#).

Usage

ig_cdash

Format

A data frame with columns:

standard Always "CDASHIG".

version Guide version, e.g. "2.3".

class General observation class, e.g. "Findings".

domain Domain, e.g. "LB".

scenario The data collection scenario or implementation option the row belongs to (e.g. "Local Processing"), or NA for variables common to the domain.

order Position within the domain and scenario, as published.

variable CDASHIG variable name, e.g. "LBORRES".

label Variable label. Not published in version 1.1 (NA for all of its rows); at most 40 characters except for 71 labels in version 2.0, which the guide publishes longer.

question_text The CRF question text the guide suggests.

prompt The CRF field prompt the guide suggests.

type As published: "Char", "Num", "Date (dd-MON-yyyy)", or "Time (24 hour)".

core CDASHIG Core designation, as published: "HR" (highly recommended), "R/C" (recommended or conditional), or "O" (optional).

sdtmig_target The SDTMIG variable the collected value maps to.

codelist_code The CDISC CT codelist C-code(s) or subset code(s) the variable uses; NA when none.

codelist_submission_value The codelist submission value, where the export gives one.

Details

A (standard, version, domain, scenario, variable) is unique.

Source

CDISC Library CSV exports, downloaded under CDISC's terms and not redistributed; see data-raw/README.md.

ig_sdtm

SDTM-side implementation-guide variable metadata

Description

One row per variable of every domain of seven implementation-guide standards, at every version: SDTMIG (3.1.2, 3.1.3, 3.2, 3.3, 3.4), SDTMIG-AP (1.0), SDTMIG-MD (1.0, 1.1), SENDIG (3.0, 3.1, 3.1.1), SENDIG-AR (1.0), SENDIG-DART (1.1), and SENDIG-GeneTox (1.0); 14 standard versions in all. Built from CDISC Library CSV exports that are not part of this package or repository (see [ig_sources](#) and `data-raw/README.md`); only variable metadata is carried, not the guides' prose. Use [get_ig](#) to retrieve a standard, and [build_domain_spec](#) to build a ready-to-use PP/SUPPPP/ADPP variable spec from it. The SDTM model itself is in [model_sdtm](#).

Usage

ig_sdtm

Format

A data frame with columns:

standard The standard, e.g. "SDTMIG" or "SENDIG-AR".

version The standard's version, e.g. "3.4".

class General observation class, e.g. "Findings".

domain Domain, e.g. "PP", "LB", "SUPPQUAL".

order Position within the domain, as published.

variable Variable name, e.g. "PPTTESTCD".

label Variable label. At most 40 characters except for 22 labels the guides themselves publish longer (see the data-integrity tests).

type "Char" or "Num".

role CDISC variable role, e.g. "Topic".

core Core designation ("Req", "Exp", "Perm", "Cond", or the published "Not used").

codelist_code CDISC CT codelist C-code(s) the variable uses, as published; several are separated by "; " (e.g. PPORRESU lists PKUNIT and four normalized-unit codelists). NA when none.

codelist_submission_values The codelist submission value(s), where the export gives them (the SEND guides do; the SDTMIG exports do not, so look the code up in [get_ct](#)).

described_value_domain A described value domain such as "ISO 8601", where the variable has one rather than a codelist.

value_list A fixed list of allowed values, e.g. the domain abbreviation for DOMAIN.

Details

A (standard, version, domain, variable) is unique.

Source

CDISC Library CSV exports, downloaded under CDISC's terms and not redistributed; see `data-raw/README.md`.

ig_sources	<i>The CDISC Library exports the datasets were built from</i>
------------	---

Description

One row per export file read by `data-raw/build_ig.R`, so which standard versions are bundled, and from exactly which files, is itself data. The files are not part of this package or repository.

Usage

```
ig_sources
```

Format

A data frame with 52 rows (34 implementation-guide and model exports, 9 CDASH, 9 QRS supplements) and columns:

file The export's file name.

version_string The export's Version value, e.g. "ADaMIG MD v1.0"; NA for the QRS supplements, whose exports have none.

standard The canonical standard name parsed from the version string (for a QRS supplement, the instrument parsed from the file name).

version The version number parsed from the version string (for a QRS supplement, from the file name).

table The dataset it was loaded into: "ig_sdtm", "model_sdtm", "ig_adam", "cdash_model", "ig_cdash", or "qrs_supplement".

rows Data rows in the file, equal to the rows loaded from it.

md5 MD5 checksum of the file, to tell whether a rebuild used the same export.

Source

CDISC Library CSV exports; see `data-raw/README.md`.

list_datasets	<i>List all available datasets in cdisccdata</i>
---------------	--

Description

List all available datasets in cdisccdata

Usage

```
list_datasets()
```

Value

A data frame with one row per dataset describing its type, version range, and latest available version.

Examples

```
list_datasets()
```

model_sdtm	<i>SDTM model variable metadata</i>
------------	-------------------------------------

Description

One row per variable of the CDISC SDTM model, at every version: 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 2.0, and 2.1. Built from CDISC Library CSV exports not redistributed with this package (see [ig_sources](#)); only variable metadata is carried, not the model's descriptions, definitions, notes, or examples. Retrieve it with `get_ig("SDTM")`.

Usage

```
model_sdtm
```

Format

A data frame with columns:

standard Always "SDTM".

version Model version, e.g. "2.1".

class Observation class or dataset class, e.g. "Findings", "Trial Design".

dataset Dataset name, or NA for class-level variables.

order Position within the class/dataset, as published.

variable Variable name, e.g. "--TESTCD".

label Variable label. "--TESTCD" is published at 46 characters in model versions 1.2 to 1.6.

type "Char" or "Num".

role CDISC variable role.

described_value_domain A described value domain, e.g. "ISO 8601"; sparse before version 2.0.

variables_qualified The variable(s) this one qualifies.

usage_restrictions Usage restrictions on the variable; version 2.0 and later, NA before.

variable_code The variable's NCI C-code; version 2.0 and later, NA before.

Details

A (standard, version, class, dataset, variable) is unique. dataset is NA for the general-observation-class variables (Events, Findings, ...), which are defined once per class with a -- prefix, and set for the datasets the model defines outright (e.g. "DM").

Source

CDISC Library CSV exports, downloaded under CDISC's terms and not redistributed; see `data-raw/README.md`.

qrs_supplement	<i>CDISC QRS supplement item metadata</i>
----------------	---

Description

One row per item of each of nine questionnaire, rating-scale, and functional-test (QRS) instrument supplements: AIMS (2.0), APACHE_II (1.0), ATLAS (1.0), CGI (2.1), HAM-A (2.1), KFSS (2.0), KPS_SCALE (2.0), PGI (1.1), and SIX_MINUTE_WALK (1.0). Built from CDISC Library CSV exports not redistributed with this package (see [ig_sources](#)); only the identifiers that map an item to SDTM controlled terminology are carried, not the item text. Reach it with `get_dataset("qrs_supplement")`.

Usage

```
qrs_supplement
```

Format

A data frame with columns:

instrument The instrument, as in the file name, e.g. "HAM-A", "SIX_MINUTE_WALK".

version The supplement's version, e.g. "2.1".

item_order Position of the item within the instrument.

test_name The item's --TEST value, e.g. "AIMS01-Muscles of Facial Expression"; at most 40 characters.

testcd_codelist_code C-code of the --TESTCD codelist for the instrument.

testcd_code The item's --TESTCD term C-code.

test_codelist_code C-code of the --TEST codelist for the instrument.

test_code The item's --TEST term C-code.

response_group The response (value list) group the item uses, or NA for the few items that have none.

Details

The exports have no Version column; instrument and version are taken from the file name (HAM-A_Supplement_v2.1.csv). A (instrument, version, item_order) is unique.

Source

CDISC Library CSV exports, downloaded under CDISC's terms and not redistributed; see data-raw/README.md.

schema_path	<i>Get path to a Define-XML XSD schema file</i>
-------------	---

Description

Get path to a Define-XML XSD schema file

Usage

```
schema_path(define_version = c("2.1", "2.0"))
```

Arguments

define_version One of "2.1" or "2.0".

Value

Full file path to the primary XSD file.

Examples

```
schema_path("2.1")
schema_path("2.0")
```

stylesheet_path	<i>Get path to a Define-XML XSLT stylesheet</i>
-----------------	---

Description

Get path to a Define-XML XSLT stylesheet

Usage

```
stylesheet_path(define_version = c("2.1", "2.0"))
```

Arguments

define_version One of "2.1" or "2.0".

Value

Full file path to the XSL stylesheet.

Examples

```
stylesheet_path("2.1")  
stylesheet_path("2.0")
```

Index

* datasets

- cdash_model, [5](#)
- ct_adam, [7](#)
- ct_sdtm, [7](#)
- datasets_catalogue, [8](#)
- ig_adam, [12](#)
- ig_cdash, [13](#)
- ig_sdtm, [15](#)
- ig_sources, [16](#)
- model_sdtm, [17](#)
- qrs_supplement, [18](#)

available_ct_versions, [3](#)

build_domain_spec, [3](#), [12](#), [15](#)

cdash_model, [5](#), [9](#)

cdisdata_versions, [6](#)

ct_adam, [7](#)

ct_sdtm, [7](#), [7](#)

datasets_catalogue, [8](#)

get_cdash, [5](#), [9](#), [10](#), [13](#)

get_ct, [3](#), [4](#), [10](#), [11](#), [15](#)

get_dataset, [10](#), [10](#)

get_ig, [3](#), [4](#), [9](#), [10](#), [11](#), [12](#), [15](#), [17](#)

ig_adam, [11](#), [12](#), [12](#)

ig_cdash, [9](#), [13](#)

ig_sdtm, [11](#), [12](#), [15](#)

ig_sources, [5](#), [9](#), [12](#), [13](#), [15](#), [16](#), [17](#), [18](#)

list_datasets, [10](#), [11](#), [17](#)

model_sdtm, [11](#), [12](#), [15](#), [17](#)

qrs_supplement, [18](#)

schema_path, [19](#)

stylesheet_path, [19](#)