Package ‘twbparser’

September 30, 2025

Title Parse 'Tableau' Workbooks into Functional Data

Version 0.2.3

Description High-performance parsing of 'Tableau' workbook files into tidy data frames and dependency graphs for other visualization tools like R 'Shiny' or 'Power BI' replication.

License MIT + file LICENSE

URL https://prigasg.github.io/twbparser/,
https://github.com/PrigasG/twbparser

BugReports https://github.com/PrigasG/twbparser/issues

Depends R (>= 4.2.0)

Imports dplyr, igraph, purrr, R6, rlang, stringr, tibble, tidyr, withr, xml2

Suggests cli, covr, ggraph, knitr, lintr, magrittr, optparse, rmarkdown, spelling, styler, testthat (>= 3.0.0), tidygraph, vctrs, zip

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

RoxygenNote 7.3.2

NeedsCompilation no

Author George Arthur [aut, cre] (ORCID: <https://orcid.org/0000-0002-1975-1459>)

Maintainer George Arthur <prigasgenthian48@gmail.com>

Repository CRAN

Date/Publication 2025-09-30 08:30:02 UTC

Contents

`build_dependency_graph` *Build a field dependency graph from calculated fields*

Description

Creates a directed graph where edges point from input fields used in a formula to the calculated output field. Tokens are extracted from bracketed references like `[Table].[Field]` or `[Field]`.

Usage

```
build_dependency_graph(fields_df)
```

Arguments

`fields_df`	A data frame with at least columns `name` and `formula`.

Value

An igraph directed graph where vertices are field names and edges represent dependencies (input -> output).

Examples

```
fields <- tibble::tibble(
name = c("X_plus_Y", "Z"),
formula = c("[X] + [Y]", "[X_plus_Y] * 2")
)
g <- build_dependency_graph(fields)
```

extract_calculated_fields   *Extract calculated fields from a TWB*

Description

Finds columns that contain <calculation> nodes and returns metadata and formulas, with a heuristic flag for table calculations.

Usage

```
extract_calculated_fields(xml_doc)
```

Arguments

xml_doc	An xml2 document for a Tableau .twb.

Value

A tibble with columns:

datasource Datasource name.

name User-visible caption or cleaned internal name.

tableau_internal_name Internal Tableau name (often bracketed).

datatype Tableau datatype.

role Tableau role.

formula Calculation formula string.

calc_class Tableau calc class (often "tableau").

is_table_calc Heuristic flag for table calcs (e.g., WINDOW_, LOOKUP).

table Raw table reference.

table_clean Cleaned table name.

Examples

```
# A tiny TWB shipped with the package:
twb <- system.file("extdata", "test_for_wenjie.twb", package = "twbparser")
stopifnot(nzchar(twb), file.exists(twb))
xml <- xml2::read_xml(twb)
# Extract calculated fields
calcs <- extract_calculated_fields(xml)
head(calcs)
```

extract_columns_with_table_source

Extract columns with their source tables from a TWB

Description

Scans top-level <datasource> nodes (excluding view-specific references) and returns fields with raw names/captions, cleaned table/field names, and basic metadata.

Usage

```
extract_columns_with_table_source(xml_doc)
```

Arguments

xml_doc	An xml2 document for a Tableau .twb.

Value

A tibble with columns:

datasource Datasource name.

name Raw column name (may include brackets/qualifiers).

caption Column caption if present.

datatype Tableau datatype.

role Tableau role.

semantic_role Semantic role if present.

table Raw table reference.

table_clean Cleaned table name (no brackets/suffix).

field_clean Cleaned field name.

Examples

```
twb <- system.file("extdata", "test_for_wenjie.twb", package = "twbparser")
stopifnot(nzchar(twb), file.exists(twb))
xml <- xml2::read_xml(twb)
fields <- extract_columns_with_table_source(xml)
```

`extract_datasource_details`

Extract datasource details from a Tableau TWB

Description

Gathers runtime tables (from the object graph), merges in named connection metadata (class, caption, targets), and augments with top-level datasource definitions (field counts, connection type, location). Also returns a filtered table of parameter datasources.

Usage

```
extract_datasource_details(xml_doc)
```

Arguments

`xml_doc`	An `xml2` document for a Tableau `.twb`.

Value

A named list with:

data_sources Tibble of datasources joined with connection metadata.

parameters Tibble of parameter datasources (if present).

all_sources Same as `data_sources` (placeholder for future variants).

Examples

```
# Preferred: from a tiny .twb
twb <- system.file("extdata", "test_for_wenjie.twb", package = "twbparser")
if (nzchar(twb) && file.exists(twb)) {
xml <- xml2::read_xml(twb)
res <- extract_datasource_details(xml)
head(res$data_sources)
}

# Alternative: from a tiny .twbx
twbx <- system.file("extdata", "test_for_zip.twbx", package = "twbparser")
if (nzchar(twbx) && file.exists(twbx)) {
members <- twbx_list(twbx)
twb_member <- members$Name[grepl("\\.twb$", members$Name)][1]
if (!is.na(twb_member)) {
xml <- xml2::read_xml(unz(twbx, twb_member))
res <- extract_datasource_details(xml)
head(res$data_sources)
  }
}
```

extract_joins *Extract Tableau join clauses from* `<relation type="join">` *nodes*

Description

Handles both column-based clauses (`<clause><column/></clause>`) and expression-based predicates (`<expression op=...>`) found in TWB XML.

Usage

```
extract_joins(xml_doc)
```

Arguments

`xml_doc`	An `xml2` document for a Tableau `.twb`.

Value

A tibble with columns:

join_type Join kind (e.g., `inner`, `left`), if available.

left_table Left table name (cleaned).

left_field Left field name.

operator Predicate operator (defaults to `"="` when missing).

right_table Right table name (cleaned).

right_field Right field name.

Examples

```
twb <- system.file("extdata", "test_for_wenjie.twb", package = "twbparser")
stopifnot(nzchar(twb), file.exists(twb))
xml <- xml2::read_xml(twb)
extract_joins(xml)
```

extract_named_connections *Extract* <named-connection> *entries from a TWB*

Description

Rich, safe extraction of <named-connection> nodes and their <connection> attributes into a tidy tibble.

Usage

```
extract_named_connections(xml_doc)
```

Arguments

xml_doc	An xml2 document for a Tableau .twb.

Value

Tibble with columns like connection_id, connection_caption, connection_class, connection_target, dbname, schema, warehouse, region, filename, and location_named.

Examples

```
# Preferred: read from a tiny '.twb'
twb <- system.file("extdata", "test_for_wenjie.twb", package = "twbparser")
if (nzchar(twb) && file.exists(twb)) {
xml <- xml2::read_xml(twb)
extract_named_connections(xml)
}

# Alternative: read from a tiny '.twbx'
twbx <- system.file("extdata", "test_for_zip.twbx", package = "twbparser")
if (nzchar(twbx) && file.exists(twbx)) {
members <- twbx_list(twbx)
twb_member <- members$Name[grepl("\\.twb$", members$Name)][1]
if (!is.na(twb_member)) {
xml <- xml2::read_xml(utils::unz(twbx, twb_member))
extract_named_connections(xml)
  }
}
```

extract_parameters *Extract parameter fields from a TWB*

Description

Returns parameter columns (those with `param-domain-type`) and basic metadata, including a best-effort current value if present.

Usage

```
extract_parameters(xml_doc)
```

Arguments

`xml_doc`	An `xml2` document for a Tableau `.twb`.

Value

A tibble with columns:

datasource Datasource name.

name User-visible caption or cleaned internal name.

tableau_internal_name Internal Tableau name.

datatype Tableau datatype.

role Tableau role.

parameter_type Tableau parameter domain type.

allowable_type Underlying data-type (if present).

current_value Current value if specified.

is_parameter Always `TRUE`.

table Raw table reference.

table_clean Cleaned table name.

Examples

```
twb <- system.file("extdata", "test_for_wenjie.twb", package = "twbparser")
stopifnot(nzchar(twb), file.exists(twb))
xml <- xml2::read_xml(twb)
params <- extract_parameters(xml)
head(params)
```

extract_raw_fields *Extract non-calculated, non-parameter fields from a TWB*

Description

Returns raw columns excluding calculated fields and parameters.

Usage

```
extract_raw_fields(xml_doc)
```

Arguments

xml_doc	An xml2 document for a Tableau .twb.

Value

A tibble with columns:

datasource Datasource name.

name User-visible caption or cleaned internal name.

tableau_internal_name Internal Tableau name.

datatype Tableau datatype.

role Tableau role.

is_hidden Whether the field is hidden.

is_parameter Always FALSE.

table Raw table reference.

table_clean Cleaned table name.

Examples

```
twb <- system.file("extdata", "test_for_wenjie.twb", package = "twbparser")
stopifnot(nzchar(twb), file.exists(twb))
xml <- xml2::read_xml(twb)
raw_fields <- extract_raw_fields(xml)
head(raw_fields)
```

extract_relations *Extract all <relation> tags from a TWB*

Description

Returns a tibble of <relation> elements found in a Tableau TWB XML, with key attributes and any custom SQL text.

Usage

```
extract_relations(xml_doc)
```

Arguments

xml_doc	An xml2 document for a Tableau .twb.

Value

A tibble with columns:

name	Relation name
table	Table reference
connection	Connection ID
type	Relation type (table, join, etc.)
join	Join type if applicable
custom_sql	Inline SQL text if present

Examples

```
twb <- system.file("extdata", "test_for_wenjie.twb", package = "twbparser")
stopifnot(nzchar(twb), file.exists(twb))
xml <- xml2::read_xml(twb)
fields <- extract_columns_with_table_source(xml)
inferred <- infer_implicit_relationships(fields)
head(inferred)
```

extract_relationships *Extract modern relationships from a Tableau TWB*

Description

Parses Tableau "relationships" (introduced in 2020.2) between logical tables, including the join predicate fields and operator.

Usage

```
extract_relationships(xml_doc)
```

Arguments

`xml_doc`	An `xml2` document for a Tableau `.twb`.

Value

A tibble with columns:

`relationship_type`	Always "Relationship"
`left_table`	Left table name
`right_table`	Right table name
`left_field`	Field name on left side
`operator`	Join operator (e.g., "=")
`right_field`	Field name on right side
`left_is_calc`	Logical, whether left field is a calculation
`right_is_calc`	Logical, whether right field is a calculation

Examples

```
twb <- system.file("extdata", "test_for_wenjie.twb", package = "twbparser")
stopifnot(nzchar(twb), file.exists(twb))
xml <- xml2::read_xml(twb)
extract_relationships(xml)
```

extract_twb_from_twbx *Extract the .twb (and optionally all files) from a .twbx*

Description

Extract the .twb (and optionally all files) from a .twbx

Usage

```
extract_twb_from_twbx(
  twbx_path,
  extract_dir = file.path(tempdir(), paste0("twbx_",
    tools::file_path_sans_ext(basename(twbx_path)), "_", format(Sys.time(),
    "%Y%m%d%H%M%S"))),
  extract_all = FALSE
)
```

Arguments

`twbx_path`	Path to a `.twbx` file.
`extract_dir`	Directory to extract into (defaults to a timestamped temp dir).
`extract_all`	If `TRUE`, extract entire archive; otherwise only the largest `.twb`.

Value

List with `twb_path`, `exdir`, `twbx_path`, and `manifest` (tibble).

Examples

```
twbx <- system.file("extdata", "test_for_zip.twbx", package = "twbparser")
res  <- extract_twb_from_twbx(twbx, extract_all = FALSE)
basename(res$twb_path)
```

infer_implicit_relationships *Infer implicit relationships between tables from field metadata*

Description

Generates candidate join pairs by:

- Matching `semantic_role` across different tables.
- Matching field names (case-insensitive) across different tables.

Usage

```
infer_implicit_relationships(fields_df, max_pairs = 50000L)
```

Arguments

`fields_df` A data frame like the output of `extract_columns_with_table_source()`.

`max_pairs` Maximum number of candidate pairs to return (default 50,000).

Value

A tibble with columns:

left_table Left table name.

left_field Left field name.

right_table Right table name.

right_field Right field name.

reason Why the pair was suggested.

Examples

```
twb <- system.file("extdata", "test_for_wenjie.twb", package = "twbparser")
stopifnot(nzchar(twb), file.exists(twb))
xml <- xml2::read_xml(twb)
fields <- extract_columns_with_table_source(xml)
inferred <- infer_implicit_relationships(fields)
head(inferred)
```

`plot_dependency_graph` *Plot a field dependency graph*

Description

Draws a quick base-graphics plot of a dependency graph. Vertices that are calculated fields (present in `fields_df$name`) are drawn differently.

Usage

```
plot_dependency_graph(g, fields_df = NULL, seed = NULL)
```

Arguments

`g` An `igraph` directed graph from `build_dependency_graph()`.

`fields_df` Optional data frame with a `name` column to mark calculated outputs.

`seed` Optional integer seed to make the layout reproducible. If `NULL` (default), the function will not alter the caller's RNG state.

Value

Invisibly returns g.

Examples

```
fields <- tibble::tibble(
name = c("X_plus_Y", "Z"),
formula = c("[X] + [Y]", "[X_plus_Y] * 2")
)
g <- build_dependency_graph(fields)
plot_dependency_graph(g, fields)          # nondeterministic layout
plot_dependency_graph(g, fields, seed = 1) # deterministic layout
```

---

plot_relationship_graph

Plot a field-level relationship DAG (legacy)

Description

Uses `relationships_df` with columns `left_table`, `right_table`, `left_field`, `right_field`, and optional `operator`.

Usage

```
plot_relationship_graph(relationships_df, seed = NULL)
```

Arguments

`relationships_df`
: Data frame of field-level relationships.

`seed`
: Optional integer seed to make the layout reproducible. If `NULL` (default), the function preserves the caller's RNG state.

Value

Invisibly returns the plotted graph.

`plot_source_join_graph` *Plot a source join graph*

Description

Visualizes joins between sources. Expects `joins_df` with columns `left_table`, `right_table`, `left_field`, `right_field`. If `relationships_df` is provided (modern relationships), it will render a second graph highlighting those relationships.

Usage

```
plot_source_join_graph(joins_df, relationships_df = NULL, seed = NULL)
```

Arguments

`joins_df`	Data frame with join edges.
`relationships_df`	Optional data frame with modern relationships.
`seed`	Optional integer seed to make layouts reproducible. If `NULL` (default), the function preserves the caller's RNG state.

Value

Invisibly returns the join graph, or a list `list(joins = g, relationships = gr)` if `relationships_df` is provided.

`prettify_calculated_fields` *Add a prettified formula column to calculated fields table*

Description

Add a prettified formula column to calculated fields table

Usage

```
prettify_calculated_fields(calcs, strip_brackets = FALSE, wrap = 100L)
```

Arguments

`calcs`	tibble from extract_calculated_fields()
`strip_brackets`	logical
`wrap`	integer wrap width; default 100

Value

tibble with extra column `formula_pretty`

`tableau_formula_pretty` *Prettify a Tableau calculation formula for display*

Description

Prettify a Tableau calculation formula for display

Usage

```
tableau_formula_pretty(formula, strip_brackets = FALSE, wrap = NA_integer_)
```

Arguments

`formula`	character scalar
`strip_brackets`	logical; remove [ ] around field names (default FALSE) [ ]: R:%20
`wrap`	optional integer to hard-wrap lines (use NA to disable)

Value

character scalar (multi-line, indented)

`tbs_custom_sql_graphql` *Custom SQL (Metadata API) for a published item*

Description

Queries the Metadata (GraphQL) API for Custom SQL tables in the content graph.

Usage

```
tbs_custom_sql_graphql(
  content_id,
  base_url = Sys.getenv("TABLEAU_BASE_URL"),
  site = Sys.getenv("TABLEAU_SITE"),
  token = Sys.getenv("TABLEAU_PAT")
)
```

Arguments

`content_id`	Character. Workbook or datasource ID (GUID).
`base_url`	Character. Server/Cloud base URL (e.g., "https://...").
`site`	Character. Site contentUrl ("" for default site).
`token`	Character. REST credentials token.

Value

A tibble with columns such as `custom_sql_name`, `custom_sql_query`, `database`, `schema`. Zero rows if none.

Examples

```
tbs_custom_sql_graphql("abc-123")
```

`tbs_publish_info` *Publish info for a workbook or datasource on 'Tableau' Server/Cloud*

Description

Returns an empty tibble when credentials are missing or the item is not found.

Usage

```
tbs_publish_info(
  content_id,
  base_url = Sys.getenv("TABLEAU_BASE_URL"),
  site = Sys.getenv("TABLEAU_SITE"),
  token = Sys.getenv("TABLEAU_PAT")
)
```

Arguments

`content_id`	Character. Workbook or datasource ID (GUID).
`base_url`	Character. Server/Cloud base URL (e.g., "https://...").
`site`	Character. Site contentUrl ("" for the default site).
`token`	Character. REST credentials token (from a prior sign-in).

Value

A tibble with columns like `content_id`, `site`, `project`, `web_url`, `created_at`, `updated_at`. May be zero rows if unavailable.

Examples

```
tbs_publish_info("abc-123")
```

`TwbParser` *Tableau Workbook Parser (R6)*

Description

Initialize the parser from a `.twb` or `.twbx` path.

Return the TWBX manifest (if available).

Return TWBX extract entries.

Return TWBX image entries.

Extract files from the TWBX to disk.

Validate relationships; optionally stop on failure.

Print a one-line summary of parsed content.

Arguments

`path`	Path to a `.twb` or `.twbx` file.
`types`	Optional vector of types (e.g., `"image"`, `"extract"`).
`pattern`	Optional regex to match archive paths.
`files`	Optional explicit archive paths to extract.
`exdir`	Output directory (defaults to parser's twbx dir or tempdir()).
`error`	If `TRUE`, `stop()` when validation fails.

Format

An R6 class generator.

Details

Create a parser for Tableau `.twb` / `.twbx` files. On initialization, the parser reads the XML and precomputes relationships, joins, fields, calculated fields, inferred relationships, and datasource details. For `.twbx`, it also extracts the largest `.twb` and records a manifest.

Fields

path Path to the `.twb` or `.twbx` file on disk.

xml_doc Parsed `xml2` document of the workbook.

twbx_path Original `.twbx` path if the workbook was packaged.

twbx_dir Directory where the `.twbx` was extracted.

twbx_manifest Tibble of `.twbx` contents from `twbx_list()`.

relations Tibble of `<relation>` nodes from `extract_relations()`.

joins Tibble of join clauses from `extract_joins()`.

relationships Tibble of modern relationships from `extract_relationships()`.

inferred_relationships Tibble of inferred relationship pairs by name and role.

datasource_details List containing `data_sources`, `parameters`, and `all_sources`.

fields Tibble of raw fields with table information.

calculated_fields Tibble of calculated fields.

last_validation Result from `validate()` as list with `ok` and `issues` elements.

Methods

new(path) Create a parser from `.twb` or `.twbx` file.

get_twbx_manifest() Return `.twbx` manifest tibble.

get_twbx_extracts() Return `.twbx` extract entries.

get_twbx_images() Return `.twbx` image entries.

extract_twbx_assets(types, pattern, files, exdir) Extract files from `.twbx` archive.

get_relations() Return relations tibble.

get_joins() Return joins tibble.

get_relationships() Return modern relationships tibble.

get_inferred_relationships() Return inferred relationship pairs.

get_datasources() Return datasource details tibble.

get_parameters() Return parameters tibble.

get_datasources_all() Return all sources tibble.

get_fields() Return raw fields tibble.

get_calculated_fields(pretty = FALSE, strip_brackets = FALSE, wrap = 100L) Return calculated fields tibble. When `pretty = TRUE`, includes a `formula_pretty` column with line breaks and indentation.

validate(error = FALSE) Validate relationships. Stops execution if `error = TRUE`.

summary() Print a brief summary to console.

`twbx_extract_files` *Extract specific files from a .twbx*

Description

Extract specific files from a .twbx

Usage

```
twbx_extract_files(
  twbx_path,
  files = NULL,
  pattern = NULL,
  types = NULL,
  exdir = NULL
)
```

Arguments

`twbx_path`	Path to a `.twbx`.
`files`	Vector of archive paths to extract (optional).
`pattern`	Perl regex to match archive paths (optional).
`types`	Subset by `.twbx` entry `type` (see `twbx_list()`) (optional).
`exdir`	Output directory (defaults to temp).

Value

Tibble with `name`, `type`, and `out_path` of extracted files.

Examples

```
twbx <- system.file("extdata", "test_for_zip.twbx", package = "twbparser")
files <- twbx_extract_files(twbx, types = c("workbook"))
head(files)
```

`twbx_list` *List contents of a Tableau .twbx*

Description

List contents of a Tableau .twbx

Usage

```
twbx_list(twbx_path)
```

Arguments

`twbx_path`	Path to a `.twbx` file.

Value

Tibble with columns: `name`, `size_bytes`, `modified`, `type`.

Examples

```
twbx <- system.file("extdata", "test_for_zip.twbx", package = "twbparser")
twbx_list(twbx)
```

`twb_custom_sql` *Extract Custom SQL relations from a TWB XML*

Description

Extract Custom SQL relations from a TWB XML

Usage

```
twb_custom_sql(xml_doc)
```

Arguments

`xml_doc`	An xml2 document for a .twb

Value

tibble with relation_name, relation_type, custom_sql

`twb_initial_sql` *Extract Initial SQL statements from connections (if present)*

Description

Extract Initial SQL statements from connections (if present)

Usage

```
twb_initial_sql(xml_doc)
```

Arguments

`xml_doc`	An xml2 document for a .twb

Value

tibble with connection_id, initial_sql

`twb_published_refs` *Detect likely references to published data sources (vs embedded)*

Description

Detect likely references to published data sources (vs embedded)

Usage

```
twb_published_refs(xml_doc)
```

Arguments

`xml_doc`	An xml2 document for a .twb

Value

tibble with datasource name, caption, likely_published, hints

`validate_relationships` *Validate relationships against available datasources and fields*

Description

Checks that relationship endpoints reference known datasource tables and that the predicate fields appear somewhere in the workbook (calculated, raw, or parameter fields), using a lenient token match (e.g., `INT([GEOID]) = GEOID`).

Usage

```
validate_relationships(parser, strict = FALSE)
```

Arguments

`parser`	A `TwbParser`-like object that exposes: `get_relationships()`, `get_datasources()`, `get_fields()`, and `get_calculated_fields()`. (S3/R6 both fine.)
`strict`	Logical. Reserved for future table-scoped checks that can be overly conservative with federated sources. Currently not used.

Value

A list with:

ok `TRUE` if no issues; `FALSE` otherwise.

issues A named list of tibbles. Possible elements:

- `unknown_tables`: endpoints not found among known tables.
- `unknown_fields`: predicate fields not found in the field pool.

Examples

```
twb <- system.file("extdata", "test_for_wenjie.twb", package = "twbparser")
if (nzchar(twb) && file.exists(twb)) {
  parser <- TwbParser$new(twb)
  res <- validate_relationships(parser)
  if (!res$ok) print(res$issues)
}
```

Index