

Package: idmlr (via r-universe)

October 7, 2026

Type Package

Title Read InDesign Markup Language (IDML) Files

Version 0.1.0

Maintainer Eli Pousson <eli.pousson@gmail.com>

Description Read InDesign Markup Language (IDML) Files into R.
Additional functions to modify and write IDML files may be
added in the future.

License MIT + file LICENSE

URL <https://github.com/elipousson/idmlr>,
<https://elipousson.github.io/idmlr/>

BugReports <https://github.com/elipousson/idmlr/issues>

Imports cli, fs, purrr, readr, rlang, utils, vctrs, xml2

Suggests DBI, RSQLite

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Config/pak/sysreqs cmake make libuv1-dev libxml2-dev libx11-dev

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

Date/Publication 2024-10-01 16:49:43 UTC

RemoteUrl <https://github.com/elipousson/idmlr>

RemoteRef HEAD

RemoteSha 11d05b40129b2c0f0e949ca8c97e0de4232a8d05

Contents

idml	2
idml_contents	3
idml_list_stories	4
idml_list_styles	5
idml_meta	6
idml_resource	7
idml_spreads	8
is_list_of_xml_documents	9
read_idml	9
Index	10

idml	<i>S3 object functions for idml class objects</i>
------	---

Description

`new_idml()` creates a idml class object from a file path and contents. `validate_idml()` checks idml objects. Used by `read_idml()` to create idml objects from a file and by other functions to validate input idml objects.

Usage

```
new_idml(file, path, contents, error_call = caller_env())

validate_idml(
  idml,
  what = "idml",
  nm = c("file", "path", "contents"),
  type = "application/vnd.adobe.indesign-idml-package",
  arg = caller_arg(idml),
  error_call = caller_env()
)
```

Arguments

file	File name for input IDML file.
path	Path to location of unzipped IDML file. This is the directory where the XML files are located.
contents	A named list of xml_document objects.
error_call	The execution environment of a currently running function, e.g. caller_env(). The function will be mentioned in error messages as the source of the error. See the call argument of <code>abort()</code> for more information.
idml	An idml class object.
type	MIMETYPE value to use in validating idml objects.

arg	An argument name as a string. This argument will be mentioned in error messages as the input that is at the origin of a problem.
-----	--

idml_contents	<i>Get contents from an IDML object</i>
---------------	---

Description

`get_idml_contents()` is a helper for extracting or converting `xml_document` objects from the `idml` list object.

Usage

```
get_idml_contents(
  idml,
  dir = NULL,
  file = NULL,
  format = "xml_document",
  ns = character(),
  parent_nm = NULL,
  unique_nm = TRUE,
  type = "attr",
  ...,
  error_call = caller_env()
)

format_idml_content(
  content,
  format = "xml_document",
  ns = character(),
  parent_nm = NULL,
  unique_nm = TRUE,
  type = "attr",
  names_to = zap(),
  allow_list = TRUE,
  ...,
  error_call = caller_env()
)
```

Arguments

idml	A <code>idml</code> class object created with <code>read_idml()</code> .
dir, file	Directory and file name from IDML contents to return. If <code>file</code> is <code>NULL</code> ,
format	"list", "xml_document", or "data.frame". If "list", contents are converted with <code>xml2::as_list()</code> using the supplied <code>ns</code> parameter.

ns	Optionally, a named vector giving prefix-url pairs, as produced by <code>xml_ns()</code> . If provided, all names will be explicitly qualified with the ns prefix, i.e. if the element bar is defined in namespace foo, it will be called foo:bar. (And similarly for attributes). Default namespaces must be given an explicit name. The ns is ignored when using <code>xml_name<-()</code> and <code>xml_set_name()</code> .
parent_nm	Name of top-level parent node. If parent_nm is a character vector, <code>extract_doc_df()</code> returns a data frame list.
unique_nm	If TRUE, ensure that parent_node names are unique before using them to extract data from the xml_document. Ignored if node_names is supplied.
type	Type of data to extract from children nodes: "attr" (attributes) or "text".
...	Needed for compatibility with generic. Unused.
error_call	The execution environment of a currently running function, e.g. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the call argument of <code>abort()</code> for more information.

Value

A list, XML document, data frame, or data frame list depending on format and the input idml object.

idml_list_stories	<i>List stories from the content of an idml object</i>
-------------------	--

Description

`idml_list_stories()` lists stories and related elements from a idml object. `idml_list_story_paras()` returns just the ParagraphStyleRange element of the stories.

Usage

```
idml_list_stories(idml, format = "contents")
```

```
idml_list_story_paras(idml, ...)
```

Arguments

idml	An idml class object.
format	Format of output to return. Must be "contents"

Value

A bare named list with four elements: Stories, StoryPreference, InCopyExportOption, and ParagraphStyleRange.

Examples

```
## Not run:
if(interactive()){
  path <- system.file("idml/letter_portrait_standard.idml", package = "idmlr")

  idml_obj <- read_idml(path)

  idml_list_stories(idml_obj)
}

## End(Not run)
```

idml_list_styles	<i>List styles from an idml object</i>
------------------	--

Description

`idml_list_styles()` takes an idml object and returns a bare list with names, properties, attributes, and content from the XML nodes definition the paragraph, character, or other styles of the specified type.

Usage

```
idml_list_styles(..., type = "paragraph", format = "list")
```

Arguments

...	Arguments passed on to <code>get_idml_styles</code>
idml	A idml class object created with <code>read_idml()</code> .
parent_nm	Name of top-level parent node. If parent_nm is a character vector, <code>extract_doc_df()</code> returns a data frame list.
type	Type of data to extract from children nodes: "attr" (attributes) or "text".
format	Output format must be "list"

Value

A bare list with four elements: a named list of style attributes (names are style names), properties for the style XML nodes, attributes for the style XML nodes, and the contents of each XML node.

Examples

```
## Not run:
if(interactive()){
  path <- system.file("idml/letter_portrait_standard.idml", package = "idmlr")

  idml_obj <- read_idml(path)
```

```

    idml_list_styles(idml_obj)
  }

## End(Not run)

```

idml_meta

Get metadata from an IDML object

Description

Get metadata from an IDML object

`get_idml_meta()` extracts a metadata elements from an `idml` object in the supplied format.

Usage

```
get_idml_meta(idml, format = "xml_document", ..., error_call = caller_env())
```

Arguments

<code>idml</code>	A <code>idml</code> class object created with <code>read_idml()</code> .
<code>format</code>	"list", "xml_document", or "data.frame". If "list", contents are converted with <code>xml2::as_list()</code> using the supplied <code>ns</code> parameter.
<code>...</code>	Arguments passed on to <code>get_idml_contents</code>
<code>ns</code>	Optionally, a named vector giving prefix-url pairs, as produced by <code>xml_ns()</code> . If provided, all names will be explicitly qualified with the <code>ns</code> prefix, i.e. if the element <code>bar</code> is defined in namespace <code>foo</code> , it will be called <code>foo:bar</code> . (And similarly for attributes). Default namespaces must be given an explicit name. The <code>ns</code> is ignored when using <code>xml_name<-()</code> and <code>xml_set_name()</code> .
<code>parent_nm</code>	Name of top-level parent node. If <code>parent_nm</code> is a character vector, <code>extract_doc_df()</code> returns a data frame list.
<code>unique_nm</code>	If TRUE, ensure that parent_node names are unique before using them to extract data from the <code>xml_document</code> . Ignored if <code>node_names</code> is supplied.
<code>type</code>	Type of data to extract from children nodes: "attr" (attributes) or "text".
<code>error_call</code>	The execution environment of a currently running function, e.g. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the <code>call</code> argument of <code>abort()</code> for more information.

idml_resource

*Get fonts, styles, and graphics from an IDML object***Description**

By default, if `format = "data.frame"`, `get_idml_fonts()` returns a single `data.frame` and `get_idml_styles()`, `get_idml_graphic()`, and `get_idml_preferences()` returns a list of `data.frame` objects values present in the input `idml` object. This format option is still under development and does not return all available data from the `xml_document`. Note, for the `get_idml_styles()`, the `parent_nm` argument is ignored and may be removed.

Usage

```
get_idml_resource(
  idml,
  resource,
  format = "xml_document",
  ...,
  error_call = caller_env()
)

get_idml_fonts(idml, format = "xml_document", parent_nm = NULL, ...)

get_idml_styles(
  idml,
  format = "xml_document",
  type = NULL,
  parent_nm = NULL,
  ...
)

get_idml_graphic(idml, format = "xml_document", parent_nm = NULL, ...)

get_idml_preferences(idml, format = "xml_document", parent_nm = NULL, ...)
```

Arguments

<code>idml</code>	A <code>idml</code> class object created with <code>read_idml()</code> .
<code>resource</code>	One or more resource names: "Graphic.xml", "Fonts.xml", or "Styles.xml"
<code>format</code>	"list", "xml_document", or "data.frame". If "list", contents are converted with <code>xml2::as_list()</code> using the supplied <code>ns</code> parameter.
<code>...</code>	Arguments passed on to <code>get_idml_contents</code>
	<code>ns</code> Optionally, a named vector giving prefix-url pairs, as produced by <code>xml_ns()</code> . If provided, all names will be explicitly qualified with the <code>ns</code> prefix, i.e. if the element <code>bar</code> is defined in namespace <code>foo</code> , it will be called <code>foo:bar</code> . (And similarly for attributes). Default namespaces must be given an explicit name. The <code>ns</code> is ignored when using <code>xml_name<-()</code> and <code>xml_set_name()</code> .

	unique_nm	If TRUE, ensure that parent_node names are unique before using them to extract data from the xml_document. Ignored if node_names is supplied.
error_call		The execution environment of a currently running function, e.g. caller_env(). The function will be mentioned in error messages as the source of the error. See the call argument of abort() for more information.
parent_nm		Name of top-level parent node. If parent_nm is a character vector, extract_doc_df() returns a data frame list.
type		Type of data to extract from children nodes: "attr" (attributes) or "text".

idml_spreads	<i>Get spreads from an IDML object</i>
--------------	--

Description

Get spreads from an IDML object

[get_idml_spreads\(\)](#) extracts a list of xml documents for spreads or a data frame from a idml object.

Usage

```
get_idml_spreads(
  idml,
  format = "list",
  names_to = "Spread",
  ...,
  error_call = caller_env()
)
```

Arguments

idml	A idml class object created with read_idml() .
format	"list", "xml_document", or "data.frame". If "list", contents are converted with xml2::as_list() using the supplied ns parameter.
names_to	By default, names(x) are lost. To keep them, supply a string to names_to and the names will be saved into a column with that name. If names_to is supplied and x is not named, the position of the elements will be used instead of the names.
...	Arguments passed on to get_idml_contents

ns Optionally, a named vector giving prefix-url pairs, as produced by [xml_ns\(\)](#). If provided, all names will be explicitly qualified with the ns prefix, i.e. if the element bar is defined in namespace foo, it will be called foo:bar. (And similarly for attributes). Default namespaces must be given an explicit name. The ns is ignored when using [xml_name<-\(\)](#) and [xml_set_name\(\)](#).

	parent_nm	Name of top-level parent node. If parent_nm is a character vector, extract_doc_df() returns a data frame list.
	unique_nm	If TRUE, ensure that parent_node names are unique before using them to extract data from the xml_document. Ignored if node_names is supplied.
	type	Type of data to extract from children nodes: "attr" (attributes) or "text".
error_call		The execution environment of a currently running function, e.g. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the call argument of abort() for more information.

is_list_of_xml_documents

Is the object a list of XML document objects?

Description

Is the object a list of XML document objects?

Usage

```
is_list_of_xml_documents(x)
```

read_idml

Read an InDesign Markup Language (IDML) file to a idml object

Description

[read_idml\(\)](#) takes a file path for an idml file and returns an idml object: a list containing a file name, file path, and the contents of the file as a list of xml_document objects and character vectors (for the MIMETYPE file only).

Usage

```
read_idml(file, ..., exdir = tempdir())
```

Arguments

file	Path to a file ending in a .idml file extension.
...	Additional parameters passed to xml2::read_xml()
exdir	Directory to extract files to when using utils::unzip() , Default: tempdir().

Value

A idml class object.

See Also

[xml2::read_xml\(\)](#)

Index

`abort()`, 2, 4, 6, 8, 9

`extract_doc_df()`, 4–6, 8, 9

`format_idml_content(idml_contents)`, 3

`get_idml_contents`, 6–8

`get_idml_contents(idml_contents)`, 3

`get_idml_contents()`, 3

`get_idml_fonts(idml_resource)`, 7

`get_idml_fonts()`, 7

`get_idml_graphic(idml_resource)`, 7

`get_idml_graphic()`, 7

`get_idml_meta(idml_meta)`, 6

`get_idml_meta()`, 6

`get_idml_preferences(idml_resource)`, 7

`get_idml_preferences()`, 7

`get_idml_resource(idml_resource)`, 7

`get_idml_spreads(idml_spreads)`, 8

`get_idml_spreads()`, 8

`get_idml_styles`, 5

`get_idml_styles(idml_resource)`, 7

`get_idml_styles()`, 7

`idml`, 2

`idml_contents`, 3

`idml_list_stories`, 4

`idml_list_stories()`, 4

`idml_list_story_paras`
 (`idml_list_stories`), 4

`idml_list_story_paras()`, 4

`idml_list_styles`, 5

`idml_list_styles()`, 5

`idml_meta`, 6

`idml_resource`, 7

`idml_spreads`, 8

`is_list_of_xml_documents`, 9

`new_idml(idml)`, 2

`new_idml()`, 2

`read_idml`, 9

`read_idml()`, 2, 3, 5–9

`utils::unzip()`, 9

`validate_idml(idml)`, 2

`validate_idml()`, 2

`xml2::as_list()`, 3, 6–8

`xml2::read_xml()`, 9

`xml_ns()`, 4, 6–8

`xml_set_name()`, 4, 6–8