

# Package: rjd3providers (via r-universe)

September 10, 2024

**Type** Package

**Title** Interface to 'JDemetra+ 3.x' time series analysis software.

**Version** 3.2.3

**Description** Interface to 'JDemetra+ 3.x' (<<https://github.com/jdemetra>>) time series analysis software. It offers full access to txt, csv, xml and spreadsheets files which are meant to be read by JDemetra+ Graphical User Interface.

**Depends** R (>= 4.1.0)

**Imports** rJava (>= 1.0-6), rjd3toolkit (>= 3.2.2)

**Remotes** github::rjdverse/rjd3toolkit@\*release

**SystemRequirements** Java (>= 17)

**License** EUPL

**URL** <https://github.com/rjdverse/rjd3providers>,  
<https://rjdverse.github.io/rjd3providers/>

**LazyData** TRUE

**Suggests** knitr, rmarkdown

**RoxygenNote** 7.3.1

**BugReports** <https://github.com/rjdverse/rjd3providers/issues>

**Encoding** UTF-8

**Collate** 'providers.R' 'jd3spreadsheet.R' 'jd3txt.R' 'jd3xml.R'  
'update\_path.R' 'zzz.R'

**Repository** <https://rjdverse.r-universe.dev>

**RemoteUrl** <https://github.com/rjdverse/rjd3providers>

**RemoteRef** v3.2.3

**RemoteSha** 57e3d5c5649f03309bb7dbac12f82e3663de2bdb

## Contents

.spreadsheet_moniker . . . . .	2
check_information . . . . .	3
set_spreadsheet_paths . . . . .	4
set_txt_paths . . . . .	4
set_xml_paths . . . . .	4
spreadsheet_change_file . . . . .	5
spreadsheet_content . . . . .	5
spreadsheet_data . . . . .	6
spreadsheet_id_properties . . . . .	6
spreadsheet_name . . . . .	7
spreadsheet_series . . . . .	7
spreadsheet_to_id . . . . .	8
spreadsheet_update_path . . . . .	8
txt_change_file . . . . .	9
txt_content . . . . .	9
txt_data . . . . .	10
txt_id_properties . . . . .	11
txt_name . . . . .	11
txt_series . . . . .	12
txt_to_id . . . . .	12
txt_update_path . . . . .	13
xml_change_file . . . . .	14
xml_content . . . . .	14
xml_data . . . . .	15
xml_id_properties . . . . .	15
xml_name . . . . .	16
xml_series . . . . .	16
xml_to_id . . . . .	17
<b>Index</b>	<b>18</b>

---

.spreadsheet\_moniker *Generates a java moniker for the corresponding id*

---

### Description

Generates a java moniker for the corresponding id

Generates a java moniker for the corresponding id

### Usage

.spreadsheet\_moniker(id)

.spreadsheet\_moniker(id)

**Arguments**

id

---

check_information	<i>Check existing JD+ object</i>
-------------------	----------------------------------

---

**Description**

Check existing JD+ object

**Usage**

```
check_information(jws, idx_sap = NULL, idx_sai = NULL)
```

**Arguments**

jws	the workspace object
idx_sap	the index (or the indices) of the SA-Processing (s) to check
idx_sai	the index (or the indices) of the SA-Item(s) to check.

**Details**

If the object idx\_sai is NULL then the function will only check if the workspace contains a SA-Processing at the index idx\_sap. If the object idx\_sap is NULL then the function will check if every SA-Processing of the workspace contains a SA-Item at the index idx\_sai. If the object idx\_sap is NULL and idx\_sai is NULL then the function will check nothing.

If the object idx\_sap and / or idx\_sai have a length > 1 then the checks are iterated over all the indices.

**Value**

This function return either a boolean (TRUE) if the SAI and the SAP exist in the WS, or an error specifying the not found object.

**Examples**

```
# library("rjd3workspace")
# ws <- rjd3workspace::.jws_open(file = "ws_production.xml")
#
# # Check if the SA-Item n°3 in the SA-Processing n°1 exist
# check_information(jws = ws, idx_sap = 1, idx_sai = 3)
#
# # Check if the SA-Items 1, 2 and 5 in the SA-Processing n°1 exist
# check_information(jws = ws, idx_sap = 1, idx_sai = c(1, 2, 5))
```

---

set\_spreadsheet\_paths *Title*

---

**Description**

Title

**Usage**

```
set_spreadsheet_paths(paths)
```

**Arguments**

paths

---

set\_txt\_paths *Title*

---

**Description**

Title

**Usage**

```
set_txt_paths(paths)
```

**Arguments**

paths

---

set\_xml\_paths *Set the paths of the provider*

---

**Description**

Set the paths of the provider

**Usage**

```
set_xml_paths(paths)
```

**Arguments**

paths

**Examples**

```
set_xml_paths(system.file("examples", package = "rjd3providers"))
```

---

spreadsheet\_change\_file

*Change the file of a moniker*

---

**Description**

Change the file of a moniker

**Usage**

```
spreadsheet_change_file(id, nfile, ofile = NULL)
```

**Arguments**

id	Identifier of the series (from its moniker)
nfile	New file name
ofile	Old file name. NULL or "" to change any file to the new file

---

spreadsheet\_content    *Title*

---

**Description**

Title

**Usage**

```
spreadsheet_content(file)
```

**Arguments**

file

spreadsheet\_data      *Title*

---

### **Description**

Title

### **Usage**

```
spreadsheet_data(  
  file,  
  sheet = 1,  
  period = 0,  
  aggregation = c("None", "Sum", "Average", "First", "Last", "Max", "Min"),  
  partialAggregation = FALSE,  
  cleanMissings = TRUE,  
  fullNames = FALSE  
)
```

### **Arguments**

fullNames

### **Examples**

```
set_spreadsheet_paths(system.file("examples", package = "rjd3providers"))  
xls_all <- spreadsheet_data("Insee.xlsx", 1)
```

---

spreadsheet\_id\_properties

*Gets the list of the properties corresponding to the identifier of a moniker*

---

### **Description**

Gets the list of the properties corresponding to the identifier of a moniker

### **Usage**

```
spreadsheet_id_properties(id)
```

### **Arguments**

id

---

spreadsheet_name	<i>Title</i>
------------------	--------------

---

**Description**

Title

**Usage**

```
spreadsheet_name()
```

---

spreadsheet_series	<i>Title</i>
--------------------	--------------

---

**Description**

Title

**Usage**

```
spreadsheet_series(  
  file,  
  sheet = 1,  
  series = 1,  
  period = 0,  
  aggregation = c("None", "Sum", "Average", "First", "Last", "Max", "Min"),  
  partialAggregation = FALSE,  
  cleanMissings = TRUE,  
  fullName = TRUE  
)
```

**Arguments**

fullName

---

spreadsheet\_to\_id      *Generates the id corresponding to a list of properties*

---

### **Description**

Generates the id corresponding to a list of properties

### **Usage**

spreadsheet\_to\_id(props)

### **Arguments**

props

---

spreadsheet\_update\_path  
*Update the path of a spreadsheet in a workspace*

---

### **Description**

Update the path of a spreadsheet in a workspace

### **Usage**

spreadsheet\_update\_path(jws, new\_path, idx\_sap = NULL, idx\_sai = NULL)

### **Arguments**

jws	the workspace object
new_path	the new raw data path of the spreadsheet
idx_sap	the index (or the indices) of the SA-Processing (s) to check
idx_sai	the index (or the indices) of the SA-Item(s) to check.

### **Details**

The spreadsheet file must be a .xlsx file. .xls file are no longer accepted.

### **Value**

This function return either NULL if the updating were successfull, or an error.



**Examples**

```
# library("rjd3workspace")
# ws <- rjd3workspace::.jws_open(file = "ws_production.xml")
#
# # Update all the second SA-Processing with a new path
# spreadsheet_update_path(
#   jws = jws_object,
#   new_path = normalizePath("../data/IPI_nace4.xlsx", mustWork = TRUE),
#   idx_sap = 2L
# )
```

---

txt_change_file	<i>Title</i>
-----------------	--------------

---

**Description**

Title

**Usage**

```
txt_change_file(id, nfile, ofile = NULL)
```

**Arguments**

id	Identifier of the series (from its moniker)
nfile	New file name
ofile	Old file name. NULL or "" to change any file to the new file

---

txt_content	<i>Title</i>
-------------	--------------

---

**Description**

Title

**Usage**

```
txt_content(
  file,
  fmt.locale = NULL,
  fmt.date = NULL,
  fmt.number = NULL,
  fmt.ignoreNumberGrouping = TRUE,
  gathering.period = 0,
```

```

gathering.aggregation = c("None", "Sum", "Average", "First", "Last", "Max", "Min"),
gathering.partialAggregation = FALSE,
gathering.includeMissing = TRUE,
charset = NULL,
delimiter = c("TAB", "SEMICOLON", "COMMA", "SPACE"),
txtQualifier = c("NONE", "QUOTE", "DOUBLE_QUOTE"),
header = TRUE,
skip = 0
)

```

### Arguments

skip

### Examples

```

set_txt_paths(system.file("examples", package = "rjd3providers"))
txt_all <- txt_content("ABS.csv", delimiter = "COMMA")

```

---

txt\_data

*Title*

---

### Description

Title

### Usage

```

txt_data(
  file,
  fmt.locale = NULL,
  fmt.date = NULL,
  fmt.number = NULL,
  fmt.ignoreNumberGrouping = TRUE,
  gathering.period = 0,
  gathering.aggregation = c("None", "Sum", "Average", "First", "Last", "Max", "Min"),
  gathering.partialAggregation = FALSE,
  gathering.includeMissing = TRUE,
  charset = NULL,
  delimiter = c("TAB", "SEMICOLON", "COMMA", "SPACE"),
  txtQualifier = c("NONE", "QUOTE", "DOUBLE_QUOTE"),
  header = TRUE,
  skip = 0
)

```

### Arguments

skip

**Examples**

```
set_txt_paths(system.file("examples", package = "rjd3providers"))
all <- txt_data("ABS.csv", delimiter = "COMMA")
```

---

txt_id_properties	<i>Gets the list of the properties corresponding to the identifier of a moniker</i>
-------------------	---

---

**Description**

Gets the list of the properties corresponding to the identifier of a moniker

**Usage**

```
txt_id_properties(id)
```

**Arguments**

id

---

txt_name	<i>Title</i>
----------	--------------

---

**Description**

Title

**Usage**

```
txt_name()
```

---

txt_series	<i>Title</i>
------------	--------------

---

**Description**

Title

**Usage**

```
txt_series(
  file,
  series,
  fmt.locale = NULL,
  fmt.date = NULL,
  fmt.number = NULL,
  fmt.ignoreNumberGrouping = TRUE,
  gathering.period = 0,
  gathering.aggregation = c("None", "Sum", "Average", "First", "Last", "Max", "Min"),
  gathering.partialAggregation = FALSE,
  gathering.includeMissing = TRUE,
  charset = NULL,
  delimiter = c("TAB", "SEMICOLON", "COMMA", "SPACE"),
  txtQualifier = c("NONE", "QUOTE", "DOUBLE_QUOTE"),
  header = TRUE,
  skip = 0
)
```

**Arguments**

skip

**Examples**

```
set_txt_paths(system.file("examples", package = "rjd3providers"))
txt_5 <- txt_series("ABS.csv", series = 15, delimiter = "COMMA")
```

---

txt_to_id	<i>Generates the id corresponding to a list of properties</i>
-----------	---

---

**Description**

Generates the id corresponding to a list of properties

**Usage**

```
txt_to_id(props)
```

**Arguments**

props

---

txt_update_path	<i>Update the path of a txt/csv file in a workspace</i>
-----------------	---

---

**Description**

Update the path of a txt/csv file in a workspace

**Usage**

```
txt_update_path(jws, new_path, idx_sap = NULL, idx_sai = NULL)
```

**Arguments**

jws	the workspace object
new_path	the new raw data path of the text data
idx_sap	the index (or the indices) of the SA-Processing (s) to check
idx_sai	the index (or the indices) of the SA-Item(s) to check.

**Value**

This function return either NULL if the updating were successfull, or an error.

**Examples**

```
# library("rjd3workspace")
# ws <- rjd3workspace::.jws_open(file = "ws_production.xml")
#
# # Update all the second SA-Processing with a new path
# spreadsheet_update_path(
#   jws = jws_object,
#   new_path = normalizePath("../data/IPI_nace4.csv", mustWork = TRUE),
#   idx_sap = 1L
# )
```

---

xml_change_file	<i>Change the file of a moniker</i>
-----------------	-------------------------------------

---

**Description**

Change the file of a moniker

**Usage**

```
xml_change_file(id, nfile, ofile = NULL)
```

**Arguments**

id	Identifier of the series (from its moniker)
nfile	New file name
ofile	Old file name. NULL or "" to change any file to the new file

---

xml_content	<i>Title</i>
-------------	--------------

---

**Description**

Title

**Usage**

```
xml_content(file, charset = NULL)
```

**Arguments**

file

**Examples**

```
set_xml_paths(system.file("examples", package = "rjd3providers"))  
xml_content("Prod.xml")
```

---

xml_data	<i>Title</i>
----------	--------------

---

**Description**

Title

**Usage**

```
xml_data(file, collection = 1, charset = NULL, fullNames = FALSE)
```

**Arguments**

fullNames

**Examples**

```
set_xml_paths(system.file("examples", package = "rjd3providers"))
xml_all <- xml_data("Prod.xml", 1, charset = "iso-8859-1")
```

---

xml_id_properties	<i>Gets the list of the properties corresponding to the identifier of a moniker</i>
-------------------	---

---

**Description**

Gets the list of the properties corresponding to the identifier of a moniker

**Usage**

```
xml_id_properties(id)
```

**Arguments**

id

**Examples**

```
set_xml_paths(system.file("examples", package = "rjd3providers"))
xml_1_5 <- xml_series("Prod.xml", 1, 5, charset = "iso-8859-1")
xml_id_properties(xml_1_5$moniker$id)
```

---

xml_name	<i>Gets the name of the provider</i>
----------	--------------------------------------

---

**Description**

Gets the name of the provider

**Usage**

```
xml_name()
```

**Examples**

```
xml_name()
```

---

xml_series	<i>Title</i>
------------	--------------

---

**Description**

Title

**Usage**

```
xml_series(file, collection = 1, series = 1, charset = NULL, fullName = TRUE)
```

**Arguments**

fullName

**Examples**

```
set_xml_paths(system.file("examples", package = "rjd3providers"))  
xml_1_5 <- xml_series("Prod.xml", 1, 5, charset = "iso-8859-1")
```



---

xml_to_id	<i>Generates the id corresponding to a list of properties</i>
-----------	---

---

**Description**

Generates the id corresponding to a list of properties

**Usage**

```
xml_to_id(props)
```

**Arguments**

props

**Examples**

```
set_xml_paths(system.file("examples", package = "rjd3providers"))
xml_1_5 <- xml_series("Prod.xml", 1, 5, charset = "iso-8859-1")
q <- xml_id_properties(xml_1_5$moniker$id)
q$series <- 50
xml_to_id(q)
```

# Index

.spreadsheet\_moniker, 2

check\_information, 3

set\_spreadsheet\_paths, 4

set\_txt\_paths, 4

set\_xml\_paths, 4

spreadsheet\_change\_file, 5

spreadsheet\_content, 5

spreadsheet\_data, 6

spreadsheet\_id\_properties, 6

spreadsheet\_name, 7

spreadsheet\_series, 7

spreadsheet\_to\_id, 8

spreadsheet\_update\_path, 8

txt\_change\_file, 9

txt\_content, 9

txt\_data, 10

txt\_id\_properties, 11

txt\_name, 11

txt\_series, 12

txt\_to\_id, 12

txt\_update\_path, 13

xml\_change\_file, 14

xml\_content, 14

xml\_data, 15

xml\_id\_properties, 15

xml\_name, 16

xml\_series, 16

xml\_to\_id, 17