Package 'RTest'

April 25, 2019

Version 1.2.4

Date 2019-04-24

Title A XML-Based Testing Framework for Automated Component Tests of R
Packages

Depends R (>= 3.1.3), testthat (>= 2.0.0), magick(>= 1.3), methods,
XML

Imports base64, jsonlite, rlang, glue, magrittr, stringr, tcltk

Suggests knitr, rmarkdown, covr

Description This provides a framework for R packages developed for
a regulatory environment. It is based on the 'testthat' unit testing system and provides the adapter
functionalities for XML-based test case definition as well as for standardized
reporting of the test results.

Collate RTest-package.R utils.R utils.examples.R utils.test.R
utils.xmlRead.R utils.xmlWrite.R RTestCollection.R RTestCase.R
RTestCase.test-adapter.R utils.testthat.R

SystemRequirements ImageMagick++: ImageMagick-c++-devel (rpm) or
libmagick++-dev (deb)

VignetteBuilder knitr

License GPL (>= 2)

RoxygenNote 6.1.1

NeedsCompilation no

Author Sebastian Wolf [aut, cre],
Matthias Pfeifer [aut, ctb],
Sergej Potapov [ctb],
Roche [cph, fnd]

Maintainer Sebastian Wolf <sebastian@mail-wolf.de>

BugReports https://github.com/zappingseb/RTest/issues

Repository CRAN

Date/Publication 2019-04-24 22:20:04 UTC

R **topics documented:**

---

arguments_creator　　　　　*Read arguments from RTest 'param'-XML Node*

Description

Read arguments from RTest 'param'-XML Node

Usage

```
arguments_creator(parameters_xml_definition, input_data = NULL)
```

Arguments

parameters_xml_definition

　　　　　　　(XMLNode) An XML Node that has elements of kind RTestData_variable, RTest-
　　　　　　　Data_image, RTestData_vector, RTestData_list, RTestData_data.frame in it. Please
　　　　　　　see the RTest.xsd to be found in file.path(find.package("RTest")/"xsd/RTest.xsd")

input_data　　(list) A named list of values of kind data.frame, character, numeric or list of
　　　　　　　those

Details

This function will read in all parameters exept the one named "RTestData_input_data" into a list
by using xmlReadData_to_list. The parameter "RTestData_input_data" is written into an additional
item of the list. The name of this item is given by the "param" attribute of the XMLNode "RTest-
Data_input_data". The value is given by the list item of the list "input_data" that can be found by
the "name" attribute of the XMLNode "RTestData_input_data".

Value

A named list. For the name of the RTestData_input_data element, please see details.

Author(s)

Sebastian Wolf <sebastian@mail-wolf.de>

---

clearTest   *Clear Test Reporter and Test Cache of Last Test Case Execution*

Description

This method resets the test information of the current test case including slots `'test.for'`, `'test.result'` and the cache in the `'tests'` test case environment.

Usage

```
## S4 method for signature 'RTestCase'
clearTest(object)
```

Arguments

object   (object) The [RTestCase-class](#) object.

Value

([RTestCase-class](#))

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

[RTestCase-class](#)

---

example_data_frame   *Function changing a data frame by adding a column*

Description

Function changing a data frame by adding a column

Usage

```
example_data_frame(data = data.frame(x = c(1, 2), y = c(1, 2)),
  mult = 1)
```

Arguments

data   (data.frame) Any data frame with numeric values

mult   (numeric) Any numeric value (length == 1)

Value

A data.frame with an additional column sum that is the rowwise sum multiplied by `mult`

Author(s)

Sebastian Wolf `<sebastian@mail-wolf.de>`

---

example_image *Function returning the Roche logo as an image at tempdir*

Description

Function returning the Roche logo as an image at tempdir

Usage

```
example_image(name = "Roche_logo.png")
```

Arguments

name `character` The name of the output image

Value

The file path to a temporar file with the given name that will contain the Roche_Logo.png that comes with RTest/images

Author(s)

Sebastian Wolf `<sebastian@mail-wolf.de>`

---

example_list *Function returning a list with three values*

Description

Function returning a list with three values

Usage

```
example_list(name_1 = "NAME1", value_2 = 1)
```

Arguments

name_1   (`character`) Name of the first list element

value_2   (`numeric`) Value of the second list element

Value

A list with three elements, a generic data frame inside the element `data.frame` a list element with the value "VALUE1" inside the element with name of parameter `name_1` and an item with the name "NAME2" and the value of `value_2` inside.

Author(s)

Sebastian Wolf `<sebastian@mail-wolf.de>`

---

example_list_large   *Function returning a list with three values and large DF*

Description

Function returning a list with three values and large DF

Usage

```
example_list_large(name_1 = "NAME1", value_2 = 1)
```

Arguments

name_1	(character) Name of the first list element
value_2	(numeric) Value of the second list element

Value

A list with three elements, a generic data frame inside the element `data.frame` a list element with the value "VALUE1" inside the element with name of parameter `name_1` and an item with the name "NAME2" and the value of `value_2` inside.

Author(s)

Sebastian Wolf `<sebastian@mail-wolf.de>`

---

example_variable   *Function returning relative difference of X and Y*

Description

Function returning relative difference of X and Y

Usage

```
example_variable(x = 1.2, y = 1)
```

Arguments

x   (numeric) X-value

y   (numeric) Y-value

Value

(X-Y)/(X)

Author(s)

Sebastian Wolf <sebastian@mail-wolf.de>

---

example_vector   *Function returning a character vector of length "rep"*

Description

Function returning a character vector of length "rep"

Usage

```
example_vector(rep = 5)
```

Arguments

rep   (numeric) Number of repetitions

Value

character vector containing rep times the word "RTest"

Author(s)

Sebastian Wolf <sebastian@mail-wolf.de>

---

exec   *Tests imported Test Cases*

Description

Tests imported Test Cases

Usage

```
## S4 method for signature 'RTestCollection'
exec(object, test.TCs = NULL,
  test.for = NULL, out.fPath = NULL, open = TRUE, ...)
```

Arguments

object	(object) The RTestCollection-class object.
test.TCs	(character) Vector with the TCs to be executed or NULL if all all TCs of the collection should be tested.
test.for	(vector(character)) Specification for which elements to test, NULL for test all elements
out.fPath	(character) Path to output file.
open	(logical) Should the generated file be opened (TRUE) or not (FALSE) after report generation.
...	(logical) Additional parameters passed to function writeExecSummary.html.

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

RTestCollection-class

Examples

```
testCollection <- new("RTestCollection",
   project.name    = "RTest Vignette",
   project.details = "Example test exectuion",
   tester          = "Example tester",
   test.start      = format(Sys.time(), "%Y-%m-%d %H:%M:%S"))

TCDir <- list.dirs(find.package("RTest"),recursive = TRUE) %>%
   grep(pattern = "xml-templates", value = TRUE)

message("Test Adapter being used is defined in Function")
message("test.RTest.test_returnValue_data.frame_cellbycell")
```

```
testCollection <- importTCsFromDir(testCollection,
    xml.dPath = TCDir[1],f.pattern  = "RTest_TC-01.xml")

testCollection <- exec(testCollection)

outf <- tempfile(fileext=".html")
writeExecSummary.html(testCollection, out.fPath = outf,open = FALSE)

stopifnot(any(grepl("passed",readLines(outf))))
```

---

execAdapter   *Execute the Adapter Function of the Test Case*

Description

This is an abstract method definition and specifies the adapter function of the individual test case adapter, which understand the test case and knows how to execute it. It has to be implemented in the specialized test classes separetly for each test project.

Usage

```
## S4 method for signature 'RTestCase'
execAdapter(object, tf.pkg, tf.pkg.i, tf.func,
  tf.func.i, out.fPathPre = NULL, ...)
```

Arguments

object	(object) The [RTestCase-class](#) object
tf.pkg	(character) The package name of the currently executed test function.
tf.pkg.i	(integer) The package iteration number of the currently executed test function.
tf.func	(character) The function name of the currently executed test function.
tf.func.i	(character) The function iteration number of the currently executed test function.
out.fPathPre	(character) Prefix incl. path to output files generated during tests
...	Additional arguments passed to the check function.

Value

(ANY) The cached result of the executed test function.

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

[RTestCase-class](#)

execCache   *Execution Cache for Test Function*

Description

Creates the execution cache for the the currently executed test function.

Usage

```
## S4 method for signature 'RTestCase'
execCache(object, tf.pkg, tf.pkg.i, tf.func,
  tf.func.i)
```

Arguments

object	(object) The [RTestCase-class](#) object
tf.pkg	(character) The package name of the currently executed test function.
tf.pkg.i	(integer) The package iteration number of the currently executed test function.
tf.func	(character) The function name of the currently executed test function.
tf.func.i	(character) The function iteration number of the currently executed test function.

Details

The execution cache includes all cached results of test functions that are part of the same package iteration and were executed prior to the current test function. The cached results will be stored in as a list() object with the following format:

```
XML:                                      execCache:
<pkg_01>
<func_01> ... </func_01>                  $funct_01
<func_01> ... </func_01>                  $funct_01[[1]]  -> result
<func_01> ... </func_01>                  $funct_01[[2]]  -> result
<func_02> ... </func_02>  <- current tf -> $funct_02       -> result
<func_03> ... </func_03>                  --- not included ---
</pkg_01>
```

Value

(list)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

[RTestCase-class](#)

expect_less_than   *Expect less or more*

Description

Expect less or more

Usage

```
expect_less_than(...)

expect_more_than(...)

expect_lt(...)

expect_gt(...)
```

Arguments

...	Any parameter sent to expect_gt or expect_lt of testthat

expect_silent_RTest   *Expect a function call to run silent*

Description

In case the function call is not silent, a message including all outputs, messages, warnings is given.

Usage

```
expect_silent_RTest(object)
```

Arguments

object	executable function call

Author(s)

Sebastian Wolf <sebastian@mail-wolf.de>

---

generic   *Generic test adapter Method*

Description

Generic test adapter Method

Arguments

object	(object) The [RTestCase-class](#) object
inputData	(list) List of input values
execCache	(list) list of already executed tests and their return values
xmlDef	(xmlNode) xmlNode of the Test case
package	(character) Name of the package to be tested
...	additional values can be given from execAdapter

Value

(list)

Author(s)

Sebastian Wolf <sebastian@mail-wolf.de>

See Also

[RTestCase-class](#)

Examples

```
options("RTest_verbose" = TRUE)

testCollection <- new("RTestCollection",
    project.name    = "RTest Vignette",
    project.details = "Example test exectuion",
    tester          = "Example tester",
    test.start      = format(Sys.time(), "%Y-%m-%d %H:%M:%S"))

TCDir <- paste0(find.package("RTest"),"/xml-templates")

testCollection <- RTest::importTCsFromDir(testCollection,
    xml.dPath = TCDir,f.pattern  = "RTest_TC-02.xml")


outf <- tempfile(fileext=".html")

funct_02 <<- function(data, mult) {   cbind(data, "sum" = apply(data, 1, sum)*mult) }
```

```
environment(funct_02) <- asNamespace('RTest')

testCollection <- RTest::exec(testCollection, out.fPath = outf, open=FALSE)
```

---

getExecDetails.html   *Create Detailed HTML Summary of the Last Execution of the Test Case*

Description

Generates a detailed summary of the last test case execution for the overall test report including all tested expectations.

Usage

```
## S4 method for signature 'RTestCase'
getExecDetails.html(object,
  report.onlyFailed = FALSE)
```

Arguments

object   (object) The [RTestCase-class](#) object.

report.onlyFailed

     (logical) Report only failed exceptions (TRUE) or all exceptions (FALSE, default).

Value

(character)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

[RTestCase-class](#)

---

getExecStates *Execution Statues of TCs*

Description

Execution Statues of TCs

Usage

```
## S4 method for signature 'RTestCollection'
getExecStates(object, test.TCs = NULL)
```

Arguments

object	(object) The [RTestCollection-class](#) object.
test.TCs	(character) Vector with the TCs to be executed or NULL if all all TCs of the collection should be tested.

Value

[character](#) Vector with test result ('failed' or 'passed') for all TCs.

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

[RTestCollection-class](#)

---

getExecSummary *Summary of the Last Execution of the Test Case*

Description

Summary of the Last Execution of the Test Case

Usage

```
## S4 method for signature 'RTestCase'
getExecSummary(object)
```

Arguments

object	(object) The [RTestCase-class](#) object.

Value

(list)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

[RTestCase-class](#)

---

getExecSummary.html   *Create HTML Summary of the Last Execution of the Test Case*

Description

Generates a summary of the last test case execution for the overall test report.

Usage

```
## S4 method for signature 'RTestCase'
getExecSummary.html(object)
```

Arguments

object   (object) The [RTestCase-class](#) object.

Value

(character)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

[RTestCase-class](#)

---

getfun   *function to derive external package functionalities*

Description

function to derive external package functionalities

Usage

```
getfun(x)
```

Arguments

x   character package :: function string

Value

functionality of the wanted function

found at https://stackoverflow.com/questions/38983179/do-call-a-function-in-r-without-loading-the-package

---

getID   *Get ID of the Test Case*

Description

Get ID of the Test Case

Usage

```
## S4 method for signature 'RTestCase'
getID(object)
```

Arguments

object   (object) The RTestCase-class object.

Value

(character)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

RTestCase-class

---

getRTM   *Get RTM for all executed test cases*

Description

This method returns the requirement traceability matrix (RTM) for all imported test cases.

Usage

```
## S4 method for signature 'RTestCollection'
getRTM(object, test.TCs = NULL, ...)
```

Arguments

object	(object) The RTestCollection-class object.
test.TCs	(character) Vector with the TCs to be executed or NULL if all all TCs of the collection should be tested.
...	Additional arguments passed to getRTMInfos.

Value

(data.frame) The RTM as data.frame table object.

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

RTestCollection-class

---

getRTMInfos   *Create RTM from executed test cases*

Description

Creates the requirement traceability matrix for this test case.

Usage

```
## S4 method for signature 'RTestCase'
getRTMInfos(object, test.for = NULL,
  cols = c("Version", "Type", "sDesc"))
```

Arguments

object	(object) The [RTestCase-class](#) object.
test.for	(character) Vector with package names that should be tested or NULL to get all packages available from the test case.
cols	(character) Defines which information columns are shown for the TCs in the RTM. The ordering in the vector is also the ordering in the output ('Version','Type','Label','sDesc','lDesc').

Value

(list) Listing of test case details (ID, Version, Type, sDesc) and assigned specification IDs (SpecIDs) and risk IDs (RiskIDs).

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

[RTestCase-class](#)

---

getRTMInMatrixShape   *Get RTM for all executed test cases in a matrix shape*

Description

This method returns the requirement traceability matrix (RTM) in matrix representation for all imported test cases of the test case collection. Thereby, the matrix can be created for the relationship function to test case or risk to test case.

Usage

```
## S4 method for signature 'RTestCollection'
getRTMInMatrixShape(object, test.TCs = NULL,
  type = "function", ...)
```

Arguments

object	(object) The [RTestCollection-class](#) object.
test.TCs	(character) Vector with the TCs to be executed or NULL if all all TCs of the collection should be tested.
type	(character) Type of the returned matrix, either 'function' or 'risk'.
...	Additional arguments passed to getRTM.

Value

(`data.frame`) The RTM as `data.frame` table object.

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

RTestCollection-class

---

getSynopsis   *Get Synopsis of the Test Case*

Description

Get Synopsis of the Test Case

Usage

```
## S4 method for signature 'RTestCase'
getSynopsis(object)
```

Arguments

object   (object) The RTestCase-class object.

Value

(`character`)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

RTestCase-class

getTC *Return Imported Test Case*

Description

Return Imported Test Case

Usage

```
## S4 method for signature 'RTestCollection'
getTC(object, tc.id)
```

Arguments

object	(object) The RTestCollection-class object
tc.id	(character) TestCase ID

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

RTestCase-class

getTestFor *Get For of Last Execution of the Test Case*

Description

Get For of Last Execution of the Test Case

Usage

```
## S4 method for signature 'RTestCase'
getTestFor(object)
```

Arguments

object	(object) The RTestCase-class object.

Value

(ANY)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

RTestCase-class

---

getTestResult *Get Result of Last Execution of the Test Case*

Description

Get Result of Last Execution of the Test Case

Usage

```
## S4 method for signature 'RTestCase'
getTestResult(object)
```

Arguments

object   (object) The RTestCase-class object.

Value

(ANY)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

RTestCase-class

---

getType   *Get Type of the Test Case*

Description

Get Type of the Test Case

Usage

```
## S4 method for signature 'RTestCase'
getType(object)
```

Arguments

object   (object) The RTestCase-class object.

Value

(character)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

RTestCase-class

---

getValidTCs   *Get a List of TCs*

Description

Get a List of TCs

Usage

```
## S4 method for signature 'RTestCollection'
getValidTCs(object, test.TCs = NULL)
```

Arguments

object   (object) The RTestCollection-class object.

test.TCs   (character) Vector with the TCs to be executed or NULL if all all TCs of the
                collection should be tested.

Value

(character) Only the valid TCs

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

RTestCollection-class

---

getXMLRoot   *Get the XML Root of the Test Case*

Description

Get the XML Root of the Test Case

Usage

```
## S4 method for signature 'RTestCase'
getXMLRoot(object)
```

Arguments

object   (object) The RTestCase-class object.

Value

(XMLNode)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

RTestCase-class

getXMLSourceFileName   *Get File Name of the XML Input File of the Test Case*

Description

Get File Name of the XML Input File of the Test Case

Usage

```
## S4 method for signature 'RTestCase'
getXMLSourceFileName(object)
```

Arguments

object   (object) The RTestCase-class object.

Value

(character)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

RTestCase-class

getXMLSourcePath   *Get Path to the XML Input File of the Test Case*

Description

Get Path to the XML Input File of the Test Case

Usage

```
## S4 method for signature 'RTestCase'
getXMLSourcePath(object)
```

Arguments

object   (object) The RTestCase-class object.

Value

(character)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

RTestCase-class

---

get_existence_of_fun   *Find out if a function is available*

Description

This function checks if a function name or method name is available in the global namespace or the desired package namespace

Usage

```
get_existence_of_fun(function_name, package)
```

Arguments

function_name	(character) The name of the function to look up
package	(character) The name of the package where this function might be hidden (not exported)

Value

"global" if it is available, "package" if it's available within the package or an Error if it is not available at all.

Author(s)

Sebastian Wolf <sebastian@mail-wolf.de>

---

htmlify_string   *function to make strings xml and html compatible*

Description

function to make strings xml and html compatible

Usage

```
htmlify_string(input_string)
```

Arguments

input_string   (character) A simple character string

Value

(character) The same string but incompatible characters are exchanged by HTML Name characters such as & for &

Author(s)

Sebastian Wolf <sebastian.wolf.sw1@roche.com>

Examples

```
input_string <- "<5"

stopifnot(htmlify_string(input_string)=="&lt;5")
```

---

importTC   *Import Test Case from XML File*

Description

This function imports a test case XML definition and adds it to the test collection.

Usage

```
## S4 method for signature 'RTestCollection'
importTC(object, xml.fPath)
```

Arguments

object	(object) The [RTestCollection-class](#) object
xml.fPath	(character) Path to XML definition file

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

[RTestCollection-class](#)

---

importTCsFromDir   *Import all Test Cases from XML Files of a Directory*

Description

This function imports a test case XML definition and adds it to the test collection.

Usage

```
## S4 method for signature 'RTestCollection'
importTCsFromDir(object, xml.dPath,
  f.pattern = "\\.xml$", f.igncase = FALSE)
```

Arguments

object	(object) The [RTestCollection-class](#) object
xml.dPath	(character) Path to directory containing the XML files
f.pattern	(character) An optional regular expression to search the input directory. Only file names which match the regular expression will be returned (passed as argument to [list.files](#)).
f.igncase	(logical) Should pattern-matching be case-insensitive or nt (passed as argument to [list.files](#)).

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

[RTestCollection-class](#)

---

initializeTests   *Initialize the Test Slot for a Test Case.*

Description

This method initializes the slots 'tests', 'test.for' and 'test.result' of a object of class 'RTestCase'. See description of RTestCase-class for further information.

Usage

```
## S4 method for signature 'RTestCase'
initializeTests(object)
```

Arguments

object   (object) The RTestCase-class object.

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

RTestCase-class

---

normalizeDate   *Reformat a Date String*

Description

This method reformats a date string for R packages as these can be very heterogenous defined in the DESCRIPTION files of packages.

Usage

```
normalizeDate(d, asDate = TRUE, months = c(jan = "january", feb =
  "februrary", mar = "march", apr = "april", may = "may", jun = "june", jul
  = "july", aug = "august", sep = "september", oct = "october", nov =
  "november", dec = "december"))
```

Arguments

d   (character) The date to be converted.

asDate   (boolean) Return as R "Date" representation (TRUE) or as character string (FALSE).

months   (object) The name of the year's month.

Value

(see Parameter asDate) Reformatted date.

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

[as.Date](#)

Examples

```
new_date <- normalizeDate("15.September.2018",FALSE)

stopifnot(new_date=="15.09.2018")
```

---

package_md5   *Function to derive an md5 hash of a package in a current session*

Description

Function to derive an md5 hash of a package in a current session

Usage

```
package_md5(package)
```

Arguments

package   (character) Name of the package to be scanned

Value

Namespace: md5 hash or tar: md5 hash. Dependent on whether the package can be loaded out
of its current Namespace (Namespace) or whether the installed binaries have to be scanned (tar) a
different md5 hash is given. Namespace packages where normally loaded using library calls

Author(s)

Sebastian Wolf <sebastian.wolf.sw1@roche.com>

Examples

```
package_md5("testthat")
```

| png2base64 | *Converts a PNG File Into a Base64 String for Using IT as Inline Image in HTML Files* |

Description

This function creates a base64 string of a PNG (e.g. png) directly into HTML via the data function.

Usage

```
png2base64(file, img.returnAsTag = FALSE, img.title = "image",
  img.width = NULL)
```

Arguments

`file`	(character) Path to PNG image.
`img.returnAsTag`	
	(boolean) TRUE, img is returned as HTML img-tag; FALSE, raw image base64 content is returned.
`img.title`	(character) Title of the HTML img-tag.
`img.width`	(character) With for HTML img-tag.

Details

The input file is read and a base64 string encoded The returned file is the value of the `img` attribute `src`.

Value

See paramter `img.returnAsTag`.

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

| quasi_capture | *Method capturing the run* |

Description

Method capturing the run

Usage

```
quasi_capture(quo, capture, label = NULL)
```

Arguments

quo	an `rlang` quo
capture	A function to derive the output / warnings / messages of the function as e.g. `evaluate_promise`
label	character A label for the evaluated value

Value

act A list including the label (`lab`), a caputre of the function (`cap`) and the code call itself as (`..`)

Author(s)

Sebastian Wolf <sebastian@mail-wolf.de>)

---

readXMLInputData   *Read Input Data of Test Case for Default XML Definitions*

Description

This functions reads the XML definitions for default RTest objects.

Usage

```
## S4 method for signature 'RTestCase'
readXMLInputData(object)
```

Arguments

object   (object) The RTestCase-class object

Details

Input datasets are contained below the XML element 'input-data' and can be used by any test packages and functions of the respective test case adapter (i.e. by the functions of the objects inheriting the class 'RTestCase'). By default major R data types are predefined under the node and can be read by this method. Thereby, the type of the imported datasets is determined by the XML element names. This method runs through all XML items, parses the item and converts it into the respective R object.
The following element types are supported:

- variableXML: variable, XSD: RTestData_variable, RTest: xmlReadData_variable

- vectorXML: vector, XSD: RTestData_vector, RTest: xmlReadData_vector

- data.frameXML: data.frame, XSD: RTestData_data.frame, RTest: xmlReadData_data.frame

- listXML: list, XSD: RTestData_list, RTest: xmlReadData_list

Value

(ANY) If multiple datasets are defined a `list` will be returned containing all datasets with the same order as in the XML file. Else, if only a single is defined, the dataset itself will be returned.

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

RTestCase-class

RTest	*RTest: A XML-based Testing System For R Packages*

Description

The R package RTest is a software framework for standardized unit testing of R packages developed and maintained by Roche-Diagnostics. It implements a general methods for loading and executing XML-based test cases as well as for reporting. RTest is not exetuable by its own as it provides only the basic and general methods for standardized testing. Therefore, extensions of the RTest package are required, which will implement the detailed and specific requirements of individual projects or packages. These extensions are the test adapters that understand the scheme of a project's test case definitions (i.e. the XML definitions) and impelement the test logic and test execution procedures. This concept allows a flexible usage of unit test framework, however, a common test system and strategy as well as report design will be maintained for all Roche-Diagnostics R-packages and for everybody using RTest.

Details

RTest uses the open source R package 'testthat' implemented by Hadley Wickham. It is a unit testing system for R and provides a set of methods for executing unit tests for checking different types of exceptions. However, it requires that the tests and exceptions are defined in the source code and does not allow a flexibile definition of input and reference values in XML files or any other file format. Therefore, it is used as unit testing system in the RTest package, which itself implements the functionalities to use XML-based test case definitions.

Author(s)

Sebastian Wolf <sebastian@mail-wolf.de>

RTest.cat *Write Text To Console*

Description

A message is written to the console if the option `RunitTestSuite_verbose` is set TRUE.

Usage

```
RTest.cat(...)
```

Arguments

`...` Passed directly to `paste0`.

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

`paste0`, `cat`

RTest.execute *Function to generally execute a Test Case collection*

Description

Function to generally execute a Test Case collection

Usage

```
RTest.execute(testcase.directory = list.dirs(find.package("RTest"),
  recursive = T) %>% grep(pattern = "xml-templates", value = T),
  f.pattern = "*.xml", project.name = "RTest Execution",
  project.details = "Example test exectuion",
  project.tester = "Example tester", report.file = tempfile(fileext =
  ".html"), ...)
```

Arguments

testcase.directory

    (character) Location of the Test Case XML files

f.pattern   (character) An additional pattern to just search for specific files with certain names

project.name   (character) Name of the project mentioned in all cover pages

project.details

    (character) Description of the project

project.tester   (character) Name of the Test executer

report.file   (character) Output file where to store the report

...   Additional arguments handed over to the exec,RTestCollection-method method

Value

No return value, but the command line output will show where to fund the report. Using the additional argument open=TRUE will open the report directly after execution

Author(s)

Sebastian Wolf <sebastian@mail-wolf.de>

Examples

```
directory_with_tests <- list.dirs(find.package('RTest'),recursive=TRUE) %>%
grep(pattern="xml-templates",value=TRUE)

RTest::RTest.execute(
 testcase.directory = directory_with_tests[1],
open=FALSE,
 f.pattern = "RTest_TC-generic.xml"
)
```

RTest.print   *Print Text To Console*

Description

The arguments are printed if the option RTest_verbose is set TRUE.

Usage

```
RTest.print(...)
```

Arguments

...	Passed directly to [print](#).

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

[print](#)

RTestCase   *Constructor for RTestCase*

Description

Constructor for RTestCase

Usage

```
RTestCase(ID = NULL, tc.type = NULL, synopsis = NULL,
  xml.fPath = NULL, xml.root = NULL, input.data = NULL,
  tests = NULL, test.for = NULL, test.result = NULL)
```

Arguments

ID	(character) ID of the TC.
tc.type	(character) Type of the TC (i.e. the class name).
synopsis	(list) Synopsis information of the TC (as defined in the XSD ComplexType 'RTestSynopis').
xml.fPath	(character) Path to XML definition file of the TC.
xml.root	(XMLNode) The imported TC definition as XMLNode-class object.
input.data	(list) The input data of the test case, which will be filled using the method [readXMLInputData](#).
tests	(list) A list with the test results or NULL, if the test case has not been executed (see 'Details').
test.for	(character) Names of the packages, which were tested in the last execution of the test case. NULL, if the test case has not been executed.
test.result	(character) Result of the last test case execution ('SUCCESS' or 'FAILURE'). NULL, if the test case has not been executed.

Value

(.Object) [RTestCase-class](#) Object

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

Examples

```
library(RTest)

xml.root <- XML::newXMLNode("func01")
RTest::xmlFromList(xml.root,
    list(
        params=list(mult=list(attributes=c(value="1",type="numeric"))),
        testspec=list(
execution=list(attributes=c("execution-type"="silent")),
"return-value"=list(attributes=c(
"compare-type"="equal",
"diff-type"="absolute",
"tolerance"=0.001
))
        )
    )
)
# Add the reference result to the params and testspec and read it in again
xml.root <- XML::xmlRoot(XML::xmlTreeParse(
        paste0("<root>",
            capture_output(print(xml.root[[1]])),
            capture_output(print(xml.root[[2]])),
            paste(xmlWriteData_data.frame(
                    "reference",
                    data=data.frame(x=c(1,1),y=c(2,2),sum=c(3,3)),
                    printXML=FALSE)
                ,collapse="\n"),"</root>")
    ))

# Define what to test in the first test
# Please check the function test.RTest.funct_01 to see
# how it tests the code of the function ("test_fun")
tests <- new.env()
testEntry <- list(
    "pkg"          = "RTest",                 # test description
    "pgk-iter"     = "1",
    "func"         = "funct_01",
    "func-iter"    = "1",
    "test-code"    = "RTest::test_fun",
    "test-adapter" = "RTestCase",
    "test-func"    = "test.RTest.funct_01",
    "pkg-desc"     = "no package desc",
    "func-desc"    = "Simple add func",
    "xpath"        = "/root",
    "reporter"     = NA,    # field for testthat reporter
    "result"       = NA,    # field for test result (failed/success)
    "cache"        = NA,    # field for caching test results
    "execresid"    = NA,    # field for test execution result id
```

```
    "specid"         = "0",   # field for test function id
    "riskid"         = "0"   # field for test function risk id
)

# Assign test entry to test environment
#     pkg-name  pkg-iter   func-name  func-iter
tests[["RTest"]][["1"]][["funct_01"]][["1"]] <- testEntry

# Create a TestCase Object
object <- RTestCase(
    ID="1",
    tc.type="RTestCase",
    synopsis= list(version="v1",author="Sebastian Wolf"),
    xml.fPath="",
    xml.root=xml.root,
    input.data=list("one"=
            data.frame(x=c(1,1),y=c(2,2))
    ),
    tests=tests,
    test.for="RTest",
    test.result=NA
)

# Run the test
object <- test(object,test.for="RTest")
stopifnot(object@test.result=="success")

# Run a failing test

object@input.data <- list("one"=
        data.frame(x=c(1,2),y=c(2,1))
)
object <- test(object,test.for="RTest")
stopifnot(object@test.result=="failed")
```

RTestCase-class   *The 'RTestCase' Class*

Description

Objects of the RTestCase class specify and implement the general behavoir of the XML-based test cases (TC).

Details

On basis of Based on a XML definition file (see 'RTestCase.xsd'), test cases are defined for the individual packages / projects to be tested. Thereby, this class definition provides the general outline of the individual test case classes (i.e. test case adapters), which have to implement the project- and package-specific requirements and test logic.

The slot 'tests' represents the execution environment for a test case and stores all information about the execution including test case metainformation, test result (success or failed), the testthat reporter and the test cache. It has the following format:

```
XML:                    tests:
tests
|- Pkg_1                [Pkg_1][1]
|    - funct_01         [Pkg_1][1][funct_01][1] = list( ... )
|- Pkg_1                [Pkg_1][2]
|   |- funct_01         [Pkg_1][2][funct_01][1] = list( ... )
|   |- funct_01         [Pkg_1][2][funct_01][2] = list( ... )
|    - funct_02         [Pkg_1][2][funct_02][1] = list( ... )
- Pkg_2                 [Pkg_2][1]
- funct_01          [Pkg_2][1][funct_01][1] = list( ... )
```

Slots

ID (character) ID of the TC.

tc.type (character) Type of the TC (i.e. the class name).

synopsis (list) Synopsis information of the TC (as defined in the XSD ComplexType 'RTest-Synopis').

xml.fPath (character) Path to XML definition file of the TC.

xml.root (XMLNode) The imported TC definition as XMLNode-class object.

input.data (list) The input data of the test case, which will be filled using the method readXMLInputData.

tests (list) A list with the test results or NULL, if the test case has not been executed (see 'Details').

test.for (character) Names of the packages, which were tested in the last execution of the test case. NULL, if the test case has not been executed.

test.result (character) Result of the last test case execution ('SUCCESS' or 'FAILURE'). NULL, if the test case has not been executed.

Author(s)

Matthias Pfeifer <matthiaspfeifer@gmx.net>

RTestCollection *Constructor for 'RTestCollection'*

Description

Constructor for 'RTestCollection'

Usage

```
RTestCollection(project.name = "RTest Execution",
  project.details = "Example test exectuion",
  project.tester = "Example tester")
```

Arguments

`project.name`   - Name of the test campaign

`project.details`

                - Details of the test campaign

`project.tester` - Name of the tester

Value

[RTestCollection-class](#) object

Author(s)

Sebastian Wolf

Examples

```
obj <- RTestCollection()
show(obj)
```

RTestCollection-class   *Class Definition 'RTestCollection'*

Description

Class Definition 'RTestCollection'

Slots

`project.name (character)` Name of the project.

`project.details (character)` Further details of the project.

`tester (character)` Name of the tester.

`test.start (character)` Start date of the testing project.

`collection (list)` The test case collection.

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

setTestMethod *Create and Save a Default Test Method For a Test Case Adapter*

Description

This function creates and saves a default test method for a RTest adapter with predefined arguments (see 'Details'). It will automatically create the generic as well as the method for the test case class using S4 style.

Usage

```
setTestMethod(f, signature = "RTestCase", definition = function(object,
  inputData, execCache, xmlDef, ...) NULL, where = parent.frame())
```

Arguments

f	(character) The name of the function.
signature	(character) The name of the corresponding test case adapter (i.e. the name of the class, which inherits 'RTestCase' and implements all functions associated to a test case type and specifications).
definition	(function) The method, which will be called if the signature matches the definitions (see 'Details').
where	(env) An environment where to set up the Method

Details

This function is a wrapper for standardized and simplified creation of S4-syle test case adapters required for RTest. The passed function has to be assigned to the corresponding test case class, which inherits [RTestCollection-class](#)) and represents the adapter for the respecive implementation of a test case type (i.e. XSD scheme). The passed function definition has to be in the following format with the following parameters. function(object, inputData, execCache, xmlDef, ...) { # implementation goes here }

- object(object) The [RTestCase-class](#) object.
- inputData(list) List with all input data provided in the XML test case.
- execCache(list) List with cached results determined in the predecessor test functions of a test case execution.
- xmlDef(XMLNode) An object of class XMLNode, which defines the current test function.
- ... Additional parameters passed to the individual test functions.

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

[setGeneric](#), [setMethod](#)

show   *Show Summary of RTestCollection Instance*

Description

Show Summary of RTestCollection Instance

Usage

```
## S4 method for signature 'RTestCollection'
show(object)
```

Arguments

object   (object) The RTestCollection-class object.

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

RTestCollection-class

Examples

```
location <- find.package("RTest")

testCollection <- new("RTestCollection",
   project.name    = "RTest Vignette",
   project.details = "Example test exectuion",
   tester          = "Example tester",
   test.start      = format(Sys.time(), "%Y-%m-%d %H:%M:%S"))

show(testCollection)

testCollection <- importTCsFromDir(testCollection,
   xml.dPath = paste0(location,"/xml-templates")
)

# Now one test case shall be imported
show(testCollection)
```

show,RTestCase-method   *Print Summary of the Test Case to Console*

Description

Print Summary of the Test Case to Console

Usage

```
## S4 method for signature 'RTestCase'
show(object)
```

Arguments

object   (object) The RTestCase-class object.

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

RTestCase-class

summary,RTestCollection-method
                    Execution Summary As R Data Object

Description

Execution Summary As R Data Object

Usage

```
## S4 method for signature 'RTestCollection'
summary(object, test.TCs = NULL)
```

Arguments

object   (object) The RTestCollection-class object.

test.TCs   (character) Vector with the TCs to be executed or NULL if all all TCs of the
                collection should be tested.

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

RTestCollection-class

---

systemInfo.host   *Summarize Host System Information*

Description

This method creates a tabular listing of current host system.

Usage

```
systemInfo.host()
```

Value

(data.frame) A character vector containing the R version information.

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

Sys.info

---

systemInfo.packages   *Summarize System Packages*

Description

This method creates a tabular listing of the packages, which are currently loaded and available to the system

Usage

```
systemInfo.packages(which = "loadedOnly")
```

Arguments

which   (character) Specifies, which packages to display. One of 'basePkgs', 'loadedOnly' or 'otherPkgs' (for details see sessionInfo.

Value

(data.frame) A table containing the packages' names, versions and build dates.

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

sessionInfo

---

systemInfo.RInst   *Summarize R Version Information*

Description

This method creates a tabular listing of current R version information.

Usage

```
systemInfo.RInst()
```

Value

(`character`) A character vector containing the R version information.

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

sessionInfo

---

test   *Execute Test Logic of the Test Case*

Description

This method executes the test logic for a test case in a `testthat` reporter environment.

Usage

```
## S4 method for signature 'RTestCase'
test(object, test.for = NULL,
  out.fPathPre = NULL, ...)
```

Arguments

`object`	(object) The [RTestCase-class](#) object.
`test.for`	(vector(character)) Specification for which elements to test, NULL to test for all elements
`out.fPathPre`	(character) Prefix incl. path to output files generated during tetest
`...`	Additional arguments passed to the check function.

Details

This method performs the test logic by iterating through all testgroups as defined in the TC XML definition file. For each test group it starts a seperate 'ListReporter', which is defined in the `testthat` package. Then, the different functions of a testgroup are executed. Therefore, this function calls the method execTCAdapter, which needs to be defined for each TestCase type separatly (e.g. for DSTAT, VCA, Calib, etc.). This method is the adapter function and knows how to read the test case and how to execute the functions, which should be tested. Thereby, all test results generated using [test_that](#) and the expect_* of the `testthat` package are recorded by the previously started reporter object. The generated test results are stored (slot test.result) and the test execution status set (slot test.status).

Value

([RTestCase-class](#))

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

[RTestCase-class](#)

Examples

```
location <- find.package("RTest")

TestCase <- RTestCase(xml.fPath =
  file.path(location,"xml-templates","RTest_TC-01.xml"))

result <- test(TestCase)

stopifnot(result@test.result == "success")
```

---

test.RTest.funct_01	*Test Function For Testing Function 'RTest::test_returnValue_data.frame_cellbycell'*

Description

Test Function For Testing Function 'RTest::test_returnValue_data.frame_cellbycell'

Arguments

object	(object) The [RTestCase-class](#) object
inputData	(list) List of input values
execCache	(list) list of already executed tests and their return values
xmlDef	(xmlNode) xmlNode of the Test case
...	additional values can be given from execAdapter

Value

(list)

Author(s)

Sebastian Wolf

See Also

[RTestCase-class](#)

---

test_execution	*Tests Silent Execution of an Function*

Description

Tests Silent Execution of an Function

Usage

```
test_execution(what, args, xmlTestSpec = NULL, ...)
```

Arguments

what, args	Parameters for execution of the test function (see [do.call](#)).
xmlTestSpec	(XMLNode) The XML definition of type 'RTestTest_variable'.
...	Additional parameters passed to [do.call](#).

Value

ANY result of test functin

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

do.call

Examples

```
value <- test_execution(
     "sum",
     list(x=2,y=3),
     xmlTestSpec=XML::xmlNode(
         name="execution",
         attrs=list('execution-type'="silent"))
)
stopifnot(value==5)

# Create a function that always produces warnings

sum_test <- function(...){
  warning("test")
  sum(...)
}

# Let this function run and crash, if it crashes check if the error contains "produced warnings"

tryCatch(
test_execution(
   "sum_test",
    list(x=2,y=3),
   xmlTestSpec=XML::xmlNode(name="execution",attrs=list("execution-type"="silent"))
),error=function(e){
    stopifnot(grepl("produced warnings",e))
 })
```

---

test_fun   *A simple function to Test the RTest package*

Description

A simple function to Test the RTest package

Usage

```
test_fun(dat, mult)
```

Arguments

dat	(data.frame) Any simple number dataframe with minimum one column
mult	(numeric) Any simple number

Value

A Table with the number vector + a sum of the vector multiplied by mult

Author(s)

Sebastian Wolf

Examples

```
dat <- data.frame(x=c(1,1))
mult <- 1
test_fun(dat,mult)
```

---

test_manualCheck_confirmWindow

Creates a Tcl/Tk Window for confirmation of the manual check

Description

Creates a Tcl/Tk Window for confirmation of the manual check

Usage

```
test_manualCheck_confirmWindow(openrecexp = NULL, expectedTxt = NULL)
```

Arguments

openrecexp	(function) Function to open a window with the received and expected values or NULL to hide all.
expectedTxt	(character) Shown as text for the expected vidow.

Value

(list) List with elements 'result': TRUE or FALSE depending on the user's choice and 'comment': inserted user comment.

Package dependencies

The tcltk package is required. Please install it from CRAN if not in your current library path.

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

XMLNode-class

---

test_manualCheck_file   *Creates a manual check via tcl/tk interface of created output file*

Description

Creates a manual check via tcl/tk interface of created output file

Usage

```
test_manualCheck_file(result, reference, xmlTestSpec = NULL,
  add.desc = NULL, openrecexp = NULL, openrecexp.exec = FALSE)
```

Arguments

result	(object) The result object to be tested.
reference	(object) The reference object.
xmlTestSpec	(XMLNode) The XML definition of type 'RTestTest_variable'.
add.desc	(character) Additional description added to the XML definition.
openrecexp	(code) Function to open/show reference and expected result.
openrecexp.exec	
	(logical) Specify, if the openrecexp-function should be executed by default or only on user decision (by click).

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

XMLNode-class

---

test_returnValue_any   *Generically compare two values with RTest*

Description

This function compares two value by a `test_returnValue_...` function that fits the class of the `reference` input parameter.

Usage

```
test_returnValue_any(result, reference, xmlTestSpec)
```

Arguments

`result`	(any) Any value of type character, numeric, data.frame or list (image links do not work!)
`reference`	(any) Any value of type character, numeric, data.frame or list (image links do not work!)
`xmlTestSpec`	(XMLNode) An XMLNode of type RTest_test_returnValue_...

Value

The function will not return anything but call `testthat` functions creating outputs in the reporter

Author(s)

Sebastian Wolf `<sebastian@mail-wolf.de>`

---

test_returnValue_data.frame_cellbycell
                    Tests   a   Standard   R   'data.frame'   Cell-By-Cell
                    ('RTestTest_data.frame_cellbycell')

Description

Tests a Standard R 'data.frame' Cell-By-Cell ('RTestTest_data.frame_cellbycell')

Usage

```
test_returnValue_data.frame_cellbycell(result, reference, xmlTestSpec,
  add.desc = NULL)
```

Arguments

result	(data.frame) The result data.frame to be tested
reference	(data.frame) The reference data.frame
xmlTestSpec	(XMLNode) The XML definition of type 'RTestTest_data.frame_cellbycell'
add.desc	(character) Additional description added to the XML definition.

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

XMLNode-class

---

test_returnValue_data.frame_shape

Tests a Standard R 'data.frame' by shape, rownames and colnames ('RTestTest_data.frame_shape')

Description

Tests a Standard R 'data.frame' by shape, rownames and colnames ('RTestTest_data.frame_shape')

Usage

```
test_returnValue_data.frame_shape(result, reference, xmlTestSpec,
  add.desc = NULL)
```

Arguments

result	(data.frame) The result data.frame to be tested
reference	(data.frame) The reference data.frame
xmlTestSpec	(XMLNode) The XML definition of type 'RTestTest_data.frame_cellbycell'
add.desc	(character) Additional description added to the XML definition.

Author(s)

Sebastian Wolf <sebastian.wolf.sw1@roche.com>

See Also

XMLNode-class

Examples

```
# Cleaning up

tryCatch(unloadNamespace("RTest"))
tryCatch(unloadNamespace("testthat"))
library(RTest)

# create some definition of tests

data <- '<test_df desc="Compare a value" diff-type="relative"
         compare-type="equal" tolerance="1E-6"/>'
xmlTestSpec <- XML::xmlRoot(XML::xmlParse(data,asText=TRUE))

# Create data frames

x <- data.frame(x=c(1,2,3,4),y=c(1,2,3,4))
y <- data.frame(x=c(1,2,3,4),y=c(1,2,3,4))
y_wrong_shape <- data.frame(x=c(1,2,3,4,5),y=c(1,2,3,4,5))
y_wrong_names <- data.frame(x=c(1,2,3,4),y1=c(1,2,3,4))

test_returnValue_data.frame_shape(x,y,xmlTestSpec)

# Test for shape

tryCatch(
    {test_returnValue_data.frame_shape(x,y_wrong_shape,xmlTestSpec)
    stop("test did not find difference")},
    error=function(e){
      stopifnot(grepl("rec.nrows",e))
      stopifnot(grepl("exp.nrows",e))
      stopifnot(grepl("not equal",e))
    })

# Test for column names

data <- '<test_df check_colnames="TRUE"
         desc="Compare a value" diff-type="relative"
         compare-type="equal" tolerance="1E-6"/>'
xmlTestSpec <- XML::xmlRoot(XML::xmlParse(data,asText=TRUE))

tryCatch({
    test_returnValue_data.frame_shape(x,y_wrong_names,xmlTestSpec)
    stop("test did not find difference")},
     error=function(e){
      stopifnot(grepl("rec.name",e))
      stopifnot(grepl("exp.name",e))
      stopifnot(grepl("not equal",e))
    })
```

---

test_returnValue_image

Tests an image file with ImageMagick ('RTestTest_image')

Description

Tests an image file with ImageMagick ('RTestTest_image')

Usage

```
test_returnValue_image(result, reference, xmlTestSpec, add.desc = NULL)
```

Arguments

result	(object) The result object to be tested.
reference	(object) The reference object.
xmlTestSpec	(XMLNode) The XML definition of type 'RTestTest_variable'.
add.desc	(character) Additional description added to the XML definition.

Author(s)

Sebastian Wolf <sebastian.wolf.sw1@roche.com>

See Also

XMLNode-class

Examples

```
# Cleaning up

tryCatch(unloadNamespace("RTest"))
tryCatch(unloadNamespace("testthat"))
library(RTest)

# create some definition of tests

data <- '<test_image desc="Compare a value" diff-type="relative"
         compare-type="equal" tolerance="0"/>'
xmlTestSpec <- XML::xmlRoot(XML::xmlParse(data,asText=TRUE))
location <- find.package("RTest")

# Create a test with equal images

result <- paste0(location,"/images/Roche_Logo.png")
reference <- paste0(location,"/images/Roche_Logo.png")
```

```
test_returnValue_image(result,reference,xmlTestSpec)

 # Create a test with images that are not equal

reference <- paste0(location,"/images/Roche_Logo_defect.png")
tryCatch(
   test_returnValue_image(result,reference,xmlTestSpec),
   error=function(e){
     if(!grepl("not equal to",e)){
       stop("image omparison defect, please check code")
     }
   })
```

---

test_returnValue_list_nodebynode

Tests a Standard R 'list' Node-By-Node ('RTestTest_list_nodebynode')

Description

Tests a Standard R 'list' Node-By-Node ('RTestTest_list_nodebynode')

Usage

```
test_returnValue_list_nodebynode(result, reference, xmlTestSpec,
   add.desc = NULL)
```

Arguments

result	(list) The result list to be tested
reference	(list) The reference list
xmlTestSpec	(XMLNode) The XML definition of type 'RTestTest_list_nodebynode'
add.desc	(character) Additional description added to the XML definition.

Author(s)

Sergej Potapov <sergej.potapov@roche.com>

See Also

XMLNode-class

---

test_returnValue_variable

Tests a Standard R 'variable' ('RTestTest_vector_variable')

Description

Tests a Standard R 'variable' ('RTestTest_vector_variable')

Usage

```
test_returnValue_variable(result, reference, xmlTestSpec,
  add.desc = NULL)
```

Arguments

result	(object) The result object to be tested.
reference	(object) The reference object.
xmlTestSpec	(XMLNode) The XML definition of type 'RTestTest_variable'.
add.desc	(character) Additional description added to the XML definition.

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

XMLNode-class

Examples

```
# Cleaning up

tryCatch(unloadNamespace("RTest"))
tryCatch(unloadNamespace("testthat"))
library(RTest)

data <- '<test_variable desc="Compare a value"
          diff-type="absolute" compare-type="equal" tolerance="1E-3"/>'
xmlTestSpec <- XML::xmlRoot(XML::xmlParse(data,asText=TRUE))

test_returnValue_variable(5,5,xmlTestSpec)

test_returnValue_variable(5.0001,5,xmlTestSpec)

# Compare variable with a stricter tolerance

data <- '<test_variable desc="Compare a value"
```

```
          diff-type="relative" compare-type="equal" tolerance="1E-6"/>'
xmlTestSpec <- XML::xmlRoot(XML::xmlParse(data,asText=TRUE))

tryCatch(unloadNamespace("RTest"))
tryCatch(unloadNamespace("testthat"))
library(RTest)

tryCatch(
    test_returnValue_variable(5.0001,5,xmlTestSpec),error=function(e){
      stopifnot(grepl("5.0001 not equal to 5.",e))
    })
```

---

test_returnValue_vector_elementbyelement

Tests a Standard R 'vector' Element-By-Element ('RTestTest_vector_elementbyelement')

Description

Tests a Standard R 'vector' Element-By-Element ('RTestTest_vector_elementbyelement')

Usage

```
test_returnValue_vector_elementbyelement(result, reference, xmlTestSpec,
  add.desc = NULL)
```

Arguments

result	(vector) The result vector to be tested
reference	(vector) The reference vector
xmlTestSpec	(XMLNode) The XML definition of type 'RTestTest_vector_elementbyelement'
add.desc	(character) Additional description added to the XML definition.

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

XMLNode-class

Examples

```
# Cleaning up

tryCatch(unloadNamespace("RTest"))
tryCatch(unloadNamespace("testthat"))
library(RTest)

data <- '<test_variable desc="Compare a value" diff-type="absolute" compare-type="equal"
        tolerance="1E-3"/>'
xmlTestSpec <- XML::xmlRoot(XML::xmlParse(data,asText=TRUE))

test_returnValue_vector_elementbyelement(c(5,5),c(5,5),xmlTestSpec)
test_returnValue_vector_elementbyelement(c(5,5),c(5,5.000001),xmlTestSpec)

data <- '<test_variable desc="Compare a value" diff-type="relative" compare-type="equal"
        tolerance="1E-6"/>'
xmlTestSpec <- XML::xmlRoot(XML::xmlParse(data,asText=TRUE))

tryCatch(unloadNamespace("RTest"))
tryCatch(unloadNamespace("testthat"))
library(RTest)

tryCatch(
    test_returnValue_vector_elementbyelement(c(5,5),c(5,5.0001),xmlTestSpec),
    error=function(e){
      stopifnot(grepl("5 not equal to 5.0001.",e))
     })
```

---

writeExecSummary.html   *Write Summary of Last Test Case Executions as HTML*

Description

Write Summary of Last Test Case Executions as HTML

Usage

```
## S4 method for signature 'RTestCollection'
writeExecSummary.html(object, out.fPath,
  test.TCs = NULL, open = TRUE, report.onlyFailed = FALSE,
  logo = NULL)
```

Arguments

object	(object) The RTestCollection-class object.
out.fPath	(character) Path to output file.

test.TCs	(character) Vector with the TCs to be executed or NULL if all all TCs of the collection should be tested.
open	(logical) Should the generated file be opened (TRUE) or not (FALSE) after report generation.
report.onlyFailed	
	(logical) Report only failed exceptions (TRUE) or all exceptions (FALSE, default).
logo	(character) Path to alternative logo file. To use the default logo, use NULL. To use no logo use NA.

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

RTestCollection-class

---

xmlFromList	*Creat an XML Node from a list*

Description

This function appends a list as an XML object to an item. The function allows setting attributes of XML items by using the "attributes" list name, therefore it can never write tags with the name "attributes"

Usage

```
xmlFromList(node, sublist)
```

Arguments

node	(XMLNode) A Node created by XML package
sublist	(list) Any list

Value

node (XMLNode) A node where the list is attached to the first XML Node

Author(s)

Sebastian Wolf <sebastian.wolf.sw1@roche.com>

Examples

```
root <- XML::newXMLNode("root")
li <- list(a = list(aa = 1, ab=2),
b=list(ba = 1,
                bb=list(x=4,
                        attributes=c(value=3)),
              bb= 2,
             bc =3))
xmlFromList(root,li)

# The result is an XML Node like this
#<root>
#  <a>
#    <aa>1</aa>
#    <ab>2</ab>
#  </a>
#  <b>
#    <ba>1</ba>
#    <bb value="3">
#      <x>4</x>
#    </bb>
#    <bb>2</bb>
#    <bc>3</bc>
#  </b>
#</root>
```

---

xmlRead.default   *General import function to reads XML data of different types*

Description

This function controls the import of input data set.

Usage

```
xmlRead.default(xmlItem)
```

Arguments

xmlItem   (XMLNode) Object of class XMLNode that defines the a list object and fullfills XSD
                 definition 'xmlReadData_list'.

Details

Based on the tag name of the input data definition in the XML file, the corresponding readXMLData_*
function is called, whereby * is a placeholder for the data type definition in the XML scheme. For
example, for XML definitions following the data.frame specification, a function readXMLData_data.frame <- function(
is expected, which implements the XML parser for data.frames and returns the data as R object.

Value

(data.frame) or (variable) or (vector) or a named list of all imported input datasets.

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

[XMLNode-class](#)

Examples

```
data <- '<text type="character">My text is awesome</text>'
item <- XML::xmlRoot(XML::xmlParse(data,asText=TRUE))
value <- RTest::xmlRead.default(item)
stopifnot(value=="My text is awesome")

data <- '<list><data.frame><col-defs>
<coldef name="Column1" type="character"/>
<coldef name="Column2" type="numeric"/>
</col-defs>
<row name="1"><cell>ID1</cell><cell>1</cell></row>
<row name="2"><cell>ID2</cell><cell>2.1</cell></row>
<row name="3"><cell>ID3</cell><cell>3.1</cell></row>
</data.frame></list>'
item <- XML::xmlRoot(XML::xmlParse(data,asText=TRUE))
value <- RTest::xmlRead.default(item)
stopifnot(dim(value)[1]==3)
stopifnot(dim(value)[2]==2)

data <- '<variable type="character" value="My text is awesome"/>'
item <- XML::xmlRoot(XML::xmlParse(data,asText=TRUE))
value <- RTest::xmlRead.default(item)
stopifnot(value=="My text is awesome")
```

---

xmlReadData_data.frame

Read XML Data From Type 'xmlReadData_data.frame' as R 'data.frame'

Description

Read XML Data From Type 'xmlReadData_data.frame' as R 'data.frame'

Usage

```
xmlReadData_data.frame(xmlItem, na_to_none = FALSE)
```

Arguments

xmlItem	(XMLNode) Object of class XMLNode that defines the a data frame and fullfills XSD definition 'xmlReadData_data.frame'.
na_to_none	(logical) Convert NAs to empty characters (i.e. ").

Value

(data.frame)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

XMLNode-class

Examples

```
data <- '<data.frame><col-defs>
<coldef name="Column1" type="character"/>
<coldef name="Column2" type="numeric"/>
</col-defs>
<row name="1"><cell>ID1</cell><cell>1</cell></row>
<row name="2"><cell>ID2</cell><cell>2.1</cell></row>
<row name="3"><cell>ID3</cell><cell>3.1</cell></row>
</data.frame>'
item <- XML::xmlRoot(XML::xmlParse(data,asText=TRUE))
value <- RTest::xmlReadData_data.frame(item)
stopifnot(dim(value)[1]==3)
stopifnot(dim(value)[2]==2)
```

---

xmlReadData_image	*Read XML Data From Type 'xmlReadData_image' as R Variable*

Description

Read XML Data From Type 'xmlReadData_image' as R Variable

Usage

```
xmlReadData_image(xmlItem)
```

Arguments

xmlItem	(XMLNode) Object of class XMLNode that defines the a simple variable and fullfills XSD definition 'xmlReadData_variable'.

Value

(vector)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

[XMLNode-class](#)

Examples

```
location <- find.package("RTest")
data <- paste0(
'<variable name="myvar" value="',location,
          '/images/Roche_Logo.png" type="character" image="TRUE"/>'
)
item <- XML::xmlRoot(XML::xmlParse(data,asText=TRUE))
value <- RTest::xmlReadData_image(item)
stopifnot(grepl("png",value$address))
```

---

xmlReadData_list   *Read XML Data From Type 'xmlReadData_list' as R 'list' (DUMMY)*

Description

DUMMY: IMPLEMENTATION HAS TO BE DONE!!

Usage

```
xmlReadData_list(xmlItem)
```

Arguments

xmlItem   (XMLNode) Object of class XMLNode that defines the a list object and fullfills XSD
                 definition 'xmlReadData_list'.

Value

(data.frame)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

[XMLNode-class](#)

Examples

```
data <- '<mylist>

<data.frame name="ITEM1">
<col-defs>
  <coldef name="Column1" type="character"/>
  <coldef name="Column2" type="numeric"/>
</col-defs>
<row name="1"><cell>ID1</cell><cell>1</cell></row>
<row name="2"><cell>ID2</cell><cell>2.1</cell></row>
<row name="3"><cell>ID3</cell><cell>3.1</cell></row>
</data.frame>
<vector name="ITEM2" type="numeric"><element>1</element><element>2</element></vector>

</mylist>
'
item <- XML::xmlRoot(XML::xmlParse(data,asText=TRUE))
value <- RTest::xmlReadData_list(item)
stopifnot(names(value)[1]=="ITEM1")
stopifnot(length(value[[2]])==2)
```

---

xmlReadData_text | *Read XML Data From Type 'xmlReadData_text' as R Variable*

Description

Read XML Data From Type 'xmlReadData_text' as R Variable

Usage

```
xmlReadData_text(xmlItem)
```

Arguments

xmlItem   (XMLNode) Object of class XMLNode that defines the a simple variable and fullfills XSD definition 'xmlReadData_text'.

Value

(vector)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

[XMLNode-class](#)

Examples

```
data <- '<variable type="character">My text is awesome</variable>'
item <- XML::xmlRoot(XML::xmlParse(data,asText=TRUE))
value <- RTest::xmlReadData_text(item)
stopifnot(value=="My text is awesome")
```

---

xmlReadData_to_list   *Read an unidentified List of Data Types from TestCase params*

Description

Read an unidentified List of Data Types from TestCase params

Usage

```
xmlReadData_to_list(xmlItem)
```

Arguments

xmlItem   Object of class XMLNode that defines a list of xmlTags that contain just elements
                 defined in RTest XSD (list, variable, text, data.frame, vector)

Value

args (list) All the elements named by their tag and containing the value defined in the xml

Author(s)

Sebastian Wolf <sebastian.wolf@roche.com>

Examples

```
data <- '<mylist>

<inputitem1>
<col-defs>
  <coldef name="Column1" type="character"/>
  <coldef name="Column2" type="numeric"/>
</col-defs>
<row name="1"><cell>ID1</cell><cell>1</cell></row>
<row name="2"><cell>ID2</cell><cell>2.1</cell></row>
<row name="3"><cell>ID3</cell><cell>3.1</cell></row>
</inputitem1>
<inputitem2 type="numeric"><element>1</element><element>2</element></inputitem2>
```

```
</mylist>
'
item <- XML::xmlRoot(XML::xmlParse(data,asText=TRUE))
value <- RTest::xmlReadData_to_list(item)

stopifnot(names(value)[1]=="inputitem1")
stopifnot(names(value)[2]=="inputitem2")
stopifnot(length(value[[2]])==2)
```

---

xmlReadData_variable   *Read XML Data From Type 'xmlReadData_variable' as R Variable*

Description

Read XML Data From Type 'xmlReadData_variable' as R Variable

Usage

```
xmlReadData_variable(xmlItem)
```

Arguments

xmlItem   (XMLNode) Object of class XMLNode that defines the a simple variable and fullfills
                 XSD definition 'xmlReadData_variable'.

Value

(vector)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

[XMLNode-class](#)

Examples

```
data <- '<variable name="myvar" value="4" type="numeric"/>'
item <- XML::xmlRoot(XML::xmlParse(data,asText=TRUE))
value <- RTest::xmlReadData_variable(item)
print("5 shall be the outcome")
print(value + 1)
```

---

xmlReadData_vector *Read XML Data From Type 'xmlReadData_vector' as R Vector*

Description

Read XML Data From Type 'xmlReadData_vector' as R Vector

Usage

```
xmlReadData_vector(xmlItem)
```

Arguments

xmlItem   (XMLNode) Object of class XMLNode that defines the a vector and fullfills XSD
                 definition 'xmlReadData_vector'.

Value

(vector)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

See Also

XMLNode-class

Examples

```
data <- '<testvector type="numeric"><element>1</element><element>2</element></testvector>'
item <- XML::xmlRoot(XML::xmlParse(data,asText=TRUE))
value <- RTest::xmlReadData_vector(item)
print("2 shall be the outcome")
print(length(value))
```

---

xmlWriteContext *Write the Opening (Header, Root-Tag) and Closing for a RTestCase*

Description

Write the Opening (Header, Root-Tag) and Closing for a RTestCase

Usage

```
xmlWriteContext(TCType, id, opening = TRUE, closing = TRUE,
  xsd.scheme = NULL, printXML = TRUE)
```

Arguments

`TCType`	(character) TC Type
`id`	(character) TC ID
`opening, closing`	
	(logical) Specify if the opening and/or the closing tags should be written.
`xsd.scheme`	(character) Path to XSD Scheme
`printXML`	(logical) Print output or return xml as R object

Value

(`list`) Opening [[1]] and Closing [[2]] of the Test Case

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

`xmlWriteData_data.frame`

Write a R 'data.frame' as XML Data of Type 'xmlRead-Data_data.frame'

Description

Write a R 'data.frame' as XML Data of Type 'xmlReadData_data.frame'

Usage

```
xmlWriteData_data.frame(elemname = "data", data, name = NULL,
  printXML = TRUE)
```

Arguments

`elemname`	(character) The name of the element's root tag
`data`	(data.frame) The data to write
`name`	(character) The data name.
`printXML`	(logical) Print output or return xml as R object

Value

(`character`)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

---

xmlWriteData_list *Write a R 'list' as XML Data of Type 'xmlReadData_list'*

Description

Write a R 'list' as XML Data of Type 'xmlReadData_list'

Usage

```
xmlWriteData_list(elemname = "list", data, name = NULL,
  printXML = TRUE)
```

Arguments

elemname	(character) The name of the element's root tag
data	(ANY) The list to write
name	(character) The name of the list
printXML	(logical) Print output or return xml as R object

Value

(character)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

---

xmlWriteData_variable *Write a R 'constat' as XML Data of Type 'xmlReadData_variable'*

Description

Write a R 'constat' as XML Data of Type 'xmlReadData_variable'

Usage

```
xmlWriteData_variable(elemname = "variable", data, name = NULL,
  printXML = TRUE)
```

Arguments

elemname	(character) The name of the element's root tag
data	(ANY) The variable to write
name	(character) The name of the variable
printXML	(logical) Print output or return xml as R object

Value

(character)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

---

xmlWriteData_vector   *Write a R 'vector' as XML Data of Type 'xmlReadData_vector'*

Description

Write a R 'vector' as XML Data of Type 'xmlReadData_vector'

Usage

```
xmlWriteData_vector(elemname = "vector", data, name = NULL,
  printXML = TRUE)
```

Arguments

elemname	(character) The name of the element's root tag
data	(vector) The vector data to write
name	(character) The data name.
printXML	(logical) Print output or return xml as R object

Value

(character)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

xmlWriteInputData   *Write the Input-Data section for a RTestCase*

Description

Write the Input-Data section for a RTestCase

Usage

```
xmlWriteInputData(..., printXML = TRUE)
```

Arguments

`...`	(`character`) Stuff to include in the input section
`printXML`	(`logical`) Print output or return xml as R object

Value

(`character`)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

---

xmlWriteSynopsis   *Write the Synopsis for a RTestCase*

Description

Write the Synopsis for a RTestCase

Usage

```
xmlWriteSynopsis(version, author, shortDescription = NULL,
  description = NULL, creationDate = NULL, changes = list(list(author
  = author, date = creationDate, desc = "Initial Version")),
  label = NULL, printXML = TRUE)
```

Arguments

`version`	(`dcharacter`) Version Number
`author`	(`character`) Author
`shortDescription`	
	(`character`) Short description
`description`	(`character`) Description
`creationDate`	(`character`) Creation Date
`changes`	(`list`) A listing of lists each representing one change with attributes 'author', 'date' and 'desc'
`label`	(`character`) Labels.
`printXML`	(`logical`) Print output or return xml as R object

Value

(`character`)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

---

xmlWriteTest	*Write the Test section for a RTestCase*

Description

Write the Test section for a RTestCase

Usage

```
xmlWriteTest(elemname, testdesc = NA, ..., printXML = TRUE)
```

Arguments

`elemname`	(`character`) The name of the element's root tag
`testdesc`	(`character`) The description of the test's root tag
`...`	(`character`) Stuff to include in the test section
`printXML`	(`logical`) Print output or return xml as R object

Value

(`character`)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

---

xmlWriteTestFunction   *Write the Test section for a RTestCase*

Description

Write the Test section for a RTestCase

Usage

```
xmlWriteTestFunction(elemname, testdesc = NA, execresid = NA,
  specid = NA, riskid = NA, params = "", reference = "",
  testspec = "", printXML = TRUE)
```

Arguments

elemname	(character) The name of the element's root tag
testdesc	(character) The description of the test's root tag
execresid	(character) Executed Risk ID
specid	(character) The Specification ID
riskid	(character) The Risk ID
params	(ANY) The Parameters of the function
reference	(ANY) The Reference tested against
testspec	(ANY) The test specification do calculate
printXML	(logical) Print output or return xml as R object

Value

(character)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

---

xmlWriteTests   *Write the Tests section for a RTestCase*

Description

Write the Tests section for a RTestCase

Usage

```
xmlWriteTests(..., printXML = TRUE)
```

Arguments

`...`	(character) Stuff to include in the tests section
`printXML`	(logical) Print output or return xml as R object

Value

(character)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

---

xmlWriteTestSpec   *Write the testpsec section for a RTestCase*

Description

Write the testpsec section for a RTestCase

Usage

```
xmlWriteTestSpec(..., printXML = TRUE)
```

Arguments

`...`	(character) Stuff to include in the tests section
`printXML`	(logical) Print output or return xml as R object

Value

(character)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

xmlWriteTest_data.frame_cellbycell
        Write XML Test Definition of Type 'RTestTest_data.frame_cellbycell'

Description

Write XML Test Definition of Type 'RTestTest_data.frame_cellbycell'

Usage

```
xmlWriteTest_data.frame_cellbycell(elemname = "test",
  desc = "testname", data, diff_type = "absolute", tolerance = 0,
  compare_type = "equal", printXML = TRUE)
```

Arguments

elemname	(character) The tag name of the test
desc	(character) Description
data	(data.frame) The reference data for which the test should be written.
diff_type	(character) Difference 'absolute' or 'relative' that is used for comparison.
tolerance	(ANY) Named vector with tolerances, single entry if same for all.
compare_type	(character) Comparator used in the XML spec.
printXML	(logical) Print output or return xml as R object

Value

(character)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

---

xmlWriteTest_execution
        Write XML Test Definition of Type 'RTestTest_execution'

Description

Write XML Test Definition of Type 'RTestTest_execution'

Usage

```
xmlWriteTest_execution(elemname = "execution", desc = NULL,
  executionType = "silent", printXML = TRUE)
```

Arguments

`elemname`	(character) The tag name of the test
`desc`	(character) The testname
`executionType`	(character) The execution mode to be checked (i.e. 'silent', 'warning', 'error').
`printXML`	(logical) Print output or return xml as R object

Value

(character)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

`xmlWriteTest_list_nodebynode`

Write XML Test Definition of Type 'RTestTest_list_nodebynode'

Description

Write XML Test Definition of Type 'RTestTest_list_nodebynode'

Usage

```
xmlWriteTest_list_nodebynode(elemname = "return-value",
  testname = "list_nodebynode", data = NULL, test = "absolute",
  tolerance = 0, printXML = TRUE)
```

Arguments

`elemname`	(character) The tag name of the test
`testname`	(character) The testname
`data`	(data.frame) The reference data for which the test should be written.
`test`	(ANY) Named vector with test, single entry if same for all.
`tolerance`	(ANY) Named vector with tolerances, single entry if same for all.
`printXML`	(logical) Print output or return xml as R object

Value

(character)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

xmlWriteTest_variable *Write XML Test Definition of Type 'RTestTest_variable'*

Description

Write XML Test Definition of Type 'RTestTest_variable'

Usage

```
xmlWriteTest_variable(elemname = "return-value", testname = "variable",
  test = "absolute", tolerance = 0, printXML = TRUE)
```

Arguments

elemname	(character) The tag name of the test
testname	(character) The testname
test	(ANY) Named vector with test, single entry if same for all.
tolerance	(ANY) Named vector with tolerances, single entry if same for all.
printXML	(logical) Print output or return xml as R object

Value

(character)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

xmlWriteTest_vector_elementbyelement
Write XML Test Definition of Type 'RTestTest_vector_elementbyelement'

Description

Write XML Test Definition of Type 'RTestTest_vector_elementbyelement'

Usage

```
xmlWriteTest_vector_elementbyelement(elemname = "return-value",
  testname = "vector_elementbyelement", data = NULL,
  test = "absolute", tolerance = 0, printXML = TRUE)
```

Arguments

elemname	(character) The tag name of the test
testname	(character) The testname
data	(data.frame) The reference data for which the test should be written.
test	(ANY) Named vector with test, single entry if same for all.
tolerance	(ANY) Named vector with tolerances, single entry if same for all.
printXML	(logical) Print output or return xml as R object

Value

(character)

Author(s)

Matthias Pfeifer <matthias.pfeifer@roche.com>

Index