Package: RDML7 (via r-universe)

October 22, 2026

Type Package

Title Read, Write, Validate, and Analyze Real-Time PCR Data in the 'RDML' Format

Version 1.0.2

Date 2026-10-14

Encoding UTF-8

Description Reads, writes, validates, converts, and manipulates quantitative polymerase chain reaction (qPCR) data through the Real-Time PCR Data Markup Language ('RDML') application programming interface (API), using 'S7' objects. Includes an extensible file-format registry, vendor and tabular-data importers, 'RDES' import and export, fluorescence-table helpers, semantic validation, and compatibility wrappers for the historical 'RDML' API.

License MIT + file LICENSE

URL https://github.com/kablag/RDML7

BugReports https://github.com/kablag/RDML7/issues

Depends R (>= 4.1.0)

Imports S7, checkmate (>= 1.6.2), data.table, readxl, stringr, tools, xml2 (>= 1.0), lubridate (>= 1.6.0)

Collate 'generics.R' 'package-doc.R' 'keyed-list.R' 'properties.R' 'classes-core.R' 'classes-schema.R' 'schema-docs.R' 'rdml-utils.R' 'conditions.R' 'xml-write.R' 'as-table.R' 'get-fdata.R' 'set-fdata.R' 'import-data.R' 'merge-rdmls.R' 'as-dendrogram.R' 'validate.R' 'summary.R' 'import-utils.R' 'import-roche.R' 'import-abi.R' 'import-rotorgene.R' 'import-excel.R' 'import-dtprime.R' 'import-csv.R' 'import-fqd.R' 'import-qstd.R' 'import-rdml.R' 'import-rdes.R' 'export-rdes.R' 'format-registry.R' 'legacy-docs.R' 'compat-api.R' 'rdml-edit.R' 'zzz.R'

Suggests knitr, chipPCR, reshape2, qpcR, ggplot2, htmltools, DT, MBmca, PCRedux, shiny, shinythemes, testthat, rmarkdown, plotly

VignetteBuilder knitr

Config/roxygen2/version 8.1.0

Config/testthat/edition 3

NeedsCompilation no

Author Konstantin A. Blagodatskikh [cre, aut], Stefan Roediger [ctb], Michal Burdukiewicz [ctb], Andrej-Nikolai Spiess [ctb]

Maintainer Konstantin A. Blagodatskikh <k.blag@yandex.ru>

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

Date/Publication 2026-10-22 19:20:42 UTC

RemoteUrl https://github.com/cran/RDML7

RemoteRef HEAD

RemoteSha bb8dc5a73a5c640e622ac8a0626f4643fbb4f346

Contents

annotationType 4
asDendrogram 4
asTable 5
asXML 7
buildRDMLFromFData 8
buildRDMLImport 9
cdnaSynthesisMethodType 10
commercialAssayType 11
cqDetectionMethodType 11
dataCollectionSoftwareType 12
dataType 13
detectRDMLFormat 15
documentationType 16
dpAmpCurveType 16
dpMeltingCurveType 17
dyeChemistryType 18
dyeType 19
editRDML 20
experimenterType 20
experimentType 21
getFData 22
gradientType 24
idReferenceType 25
idType 26
isValidRDML 26
labelFormatType 27
legacy-api 28
lidOpenType 30
listRDMLFormats 31

loadRDMLModule . . . 31
loopType . . . 32
measureType . . . 33
mergeRDMLs . . . 33
metadata-schema-types . . . 35
newRDMLImportData . . . 35
newRDMLImportSeries . . . 37
newRDMLLossRecord . . . 38
nucleotideType . . . 39
oligoType . . . 40
partitionDataType . . . 41
partitionsType . . . 42
pauseType . . . 43
pcrFormatType . . . 44
primingMethodType . . . 45
quantityType . . . 46
quantityUnitType . . . 47
rdml-experimental-types . . . 48
rdml-sample-target-types . . . 48
rdml-thermal-types . . . 49
rdmlIdType . . . 50
rdmlImportData . . . 51
rdmlImportSeries . . . 52
rdmlKeyedList . . . 52
rdmlType . . . 53
reactPosition . . . 54
reactType . . . 55
readRDML . . . 56
registerRDMLFormat . . . 57
runType . . . 58
sampleTargetType . . . 59
sampleType . . . 60
sampleTypeType . . . 62
sequencesType . . . 63
setFData . . . 64
stepType . . . 65
summary,RDML7::rdmlType-method . . . 66
targetType . . . 67
targetTypeType . . . 69
temperatureBaseType . . . 70
temperatureType . . . 71
templateQuantityType . . . 72
thermalCyclingConditionsType . . . 72
unregisterRDMLFormat . . . 74
validateRDML . . . 74
writeRDML . . . 75
xRefType . . . 76

Index **77**

annotationType — *annotationType S7 class*

Description

Free-form property/value annotation attached to a sample. Inherits from ‘rdmlBaseType‘.

Usage

```
annotationType(property = NA_character_, value = NA_character_)
```

Arguments

property	‘character(1)‘ or ‘NA‘. Annotation property name.
value	‘character(1)‘ or ‘NA‘. Annotation value.

Format

An S7 class.

Value

An S7 object of class ‘annotationType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘property‘ ‘character(1)‘ or ‘NA‘. Annotation property name.

‘value‘ ‘character(1)‘ or ‘NA‘. Annotation value.

See Also

‘sampleType‘

asDendrogram — *Represent RDML structure as a dendrogram*

Description

Groups stored data hierarchically by experiment, run, target, sample type, and curve type (‘adp‘ or ‘mdp‘). Leaf labels contain the number of matching reactions.

Usage

```
asDendrogram(x, ...)
```

Arguments

x	An [rdmlType] object.
...	Reserved for method-specific extensions.

Value

A base R ‘dendrogram‘ object. An empty RDML object returns an empty dendrogram; plotting it produces a warning.

Arguments for the ‘rdmlType‘ method

‘plotDendrogram‘ Logical. Plot the dendrogram immediately? Defaults to ‘TRUE‘. Use ‘FALSE‘ to build the object without opening a graphics device.

See Also

‘summary()‘

Examples

```
rdmlFile <- system.file(
  "extdata", "BioRad_qPCR_melt.rdml", package = "RDML7"
)
rdml <- readRDML(rdmlFile, showProgress = FALSE)
tree <- asDendrogram(rdml, plotDendrogram = FALSE)
tree
```

---

asTable — *Build a metadata table from an RDML object*

Description

Creates one metadata row for every target-specific ‘dataType‘ stored in an RDML object. This table can be inspected directly or supplied as ‘request‘ to [getFData()].

Usage

```
asTable(x, ...)
```

Arguments

x	An [rdmlType] object.
...	Named expressions defining additional columns. Expressions are evaluated once per ‘dataType‘; see **Custom columns** below.

Value

A keyed [data.table::data.table] with ‘fdataName‘ followed by the selected built-in and custom columns. Positions use canonical plate labels such as ‘A01‘.

Arguments for the ‘rdmlType‘ method

‘columns‘ Character vector selecting built-in columns, or ‘NULL‘ for all built-in columns. Available values are ‘expId‘, ‘runId‘, ‘reactId‘, ‘position‘, ‘sample‘, ‘target‘, ‘targetDyeId‘, ‘sampleType‘, ‘adp‘, and ‘mdp‘.

‘namePattern‘ Character template or expression used to create the unique ‘fdataName‘ for each curve. Character templates use ‘{field}‘ placeholders, for example ‘"{position}_{target}"‘.

‘includeHidden‘ Logical. Include experiments whose identifiers start with ‘.‘? Defaults to ‘FALSE‘.

‘treatNullAsNa‘ Logical. Convert ‘NULL‘ results from custom expressions to ‘NA‘? Defaults to ‘FALSE‘.

‘default‘, ‘addColumns‘ Deprecated compatibility arguments for named lists of base and additional expressions. New code should use ‘columns‘ and named expressions in ‘...‘.

Custom columns

Expressions in ‘...‘ can use the scalar values ‘expId‘, ‘runId‘, ‘reactId‘, ‘position‘, ‘sample‘, ‘target‘, ‘targetDyeId‘, ‘sampleType‘, ‘adp‘, and ‘mdp‘. They can also inspect the current ‘experiment‘, ‘run‘, ‘react‘, and ‘data‘ objects and the top-level ‘samples‘, ‘targets‘, ‘dateMade‘, ‘dateUpdated‘, ‘id‘, ‘experimenter‘, ‘documentation‘, ‘dye‘, and ‘thermalCyclingConditions‘ collections.

Named custom columns may be referenced by ‘namePattern‘ placeholders.

See Also

‘getFData()‘, ‘setFData()‘

Examples

```
rdmlFile <- system.file("extdata", "lc96_bACTXY.rdml", package = "RDML7")
rdml <- readRDML(rdmlFile, showProgress = FALSE)

meta <- asTable(
  rdml,
  columns = c("position", "sample", "target", "adp"),
  cq = data$cq,
  namePattern = "{position}_{target}"
)
head(meta)
```

asXML *Serialize an RDML object to XML*

Description

Serializes standard RDML properties. Package-only extension fields are kept in memory but are not emitted as non-standard XML elements.

Usage

```
asXML(x, ...)
```

Arguments

x	An [rdmlType] object.
...	Arguments passed to the method.

Value

Without ‘fileName‘, a character scalar containing XML. With ‘fileName‘, the serialized XML is returned invisibly after the file is written.

Arguments for the ‘rdmlType‘ method

‘fileName‘ Optional output path. If omitted, return the complete XML document as a character scalar. A ‘.xml‘ path writes plain XML; other extensions, normally ‘.rdml‘, write a ZIP-based RDML archive.

‘loss‘ Policy for package data that standard RDML XML cannot represent: ‘"warn"‘ (default), ‘"error"‘, or ‘"allow"‘.

See Also

‘writeRDML()‘

Examples

```
rdmlFile <- system.file("extdata", "lc96_bACTXY.rdml", package = "RDML7")
rdml <- readRDML(rdmlFile, showProgress = FALSE)

xml <- asXML(rdml, loss = "allow")
substr(xml, 1L, 80L)

output <- tempfile(fileext = ".xml")
invisible(asXML(rdml, fileName = output, loss = "allow"))
file.exists(output)
unlink(output)
```

buildRDMLFromFData *Create an RDML object from fluorescence and metadata tables*

Description

Create an RDML object from fluorescence and metadata tables

Usage

```
buildRDMLFromFData(
  fdata,
  description,
  fdataType = "adp",
  publisher = NULL,
  serialNumber = "1",
  ...
)
```

Arguments

fdata	Fluorescence table accepted by ‘setFData()‘.
description	CamelCase metadata table accepted by ‘setFData()‘.
fdataType	‘"adp"‘ or ‘"mdp"‘.
publisher	Optional publisher/importer identifier.
serialNumber	Top-level serial number.
...	Additional arguments forwarded to ‘buildRDMLImport()‘.

Value

‘rdmlType‘.

See Also

‘setFData‘, ‘rdmlImportData‘

Examples

```
rdmlFile <- system.file("extdata", "lc96_bACTXY.rdml", package = "RDML7")
source <- readRDML(rdmlFile, showProgress = FALSE)
description <- asTable(source)
description <- description[seq_len(min(2L, nrow(description))), ]
fluorescence <- getFData(source, request = description, dpType = "adp")

rebuilt <- buildRDMLFromFData(
  fdata = fluorescence,
  description = description,
  fdataType = "adp",
```

```
  publisher = "RDML7 example"
)
asTable(rebuilt)
```

---

buildRDMLImport *Build rdmlType from canonical importer data*

Description

Build rdmlType from canonical importer data

Usage

```
buildRDMLImport(importData, loss = c("warn", "error", "allow"), ...)
```

Arguments

importData	‘rdmlImportData‘.
loss	How declared lossy conversions are handled: ‘"warn"‘, ‘"error"‘, or ‘"allow"‘.
...	Additional arguments forwarded to ‘setFData()‘.

Value

‘rdmlType‘.

See Also

‘readRDML‘, ‘rdmlImportData‘

Examples

```
rdmlFile <- system.file("extdata", "lc96_bACTXY.rdml", package = "RDML7")
source <- readRDML(rdmlFile, showProgress = FALSE)
description <- asTable(source)
description <- description[seq_len(min(2L, nrow(description))), ]
fluorescence <- getFData(source, request = description, dpType = "adp")

series <- newRDMLImportSeries("adp", fluorescence, description)
importData <- newRDMLImportData(
  series = list(series),
  publisher = "RDML7 example",
  format = "rdml"
)
imported <- buildRDMLImport(importData, loss = "allow")
asTable(imported)
```

cdnaSynthesisMethodType

cdnaSynthesisMethodType S7 class

Description

Description of reverse-transcription/cDNA-synthesis conditions. Inherits from ‘rdmlBaseType‘.

Usage

```
cdnaSynthesisMethodType(
  enzyme = NA_character_,
  primingMethod = NA,
  dnaseTreatment = NA,
  thermalCyclingConditions = NA
)
```

Arguments

`enzyme`	‘character(1)‘ or ‘NA‘. Reverse-transcriptase/enzyme description.
`primingMethod`	‘primingMethodType‘ or ‘NA‘. Priming strategy.
`dnaseTreatment`	‘logical(1)‘ or ‘NA‘. Whether DNase treatment was performed.
`thermalCyclingConditions`	‘idReferenceType‘ or ‘NA‘. Reference to the thermal program used for cDNA synthesis.

Format

An S7 class.

Value

An S7 object of class ‘cdnaSynthesisMethodType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘enzyme‘ ‘character(1)‘ or ‘NA‘. Reverse-transcriptase/enzyme description.

‘primingMethod‘ ‘primingMethodType‘ or ‘NA‘. Priming strategy.

‘dnaseTreatment‘ ‘logical(1)‘ or ‘NA‘. Whether DNase treatment was performed.

‘thermalCyclingConditions‘ ‘idReferenceType‘ or ‘NA‘. Reference to the thermal program used for cDNA synthesis.

See Also

‘primingMethodType‘, ‘thermalCyclingConditionsType‘, ‘sampleType‘

commercialAssayType *commercialAssayType S7 class*

Description

Commercial assay identification. Inherits from ‘rdmlBaseType‘.

Usage

```
commercialAssayType(company = character(0), orderNumber = character(0))
```

Arguments

company	‘character(1)‘. Manufacturer/company.
orderNumber	‘character(1)‘. Catalogue or order number.

Format

An S7 class.

Value

An S7 object of class ‘commercialAssayType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘company‘ ‘character(1)‘. Manufacturer/company.

‘orderNumber‘ ‘character(1)‘. Catalogue or order number.

See Also

‘targetType‘

cqDetectionMethodType *cqDetectionMethodType S7 enumeration*

Description

Method used to determine Cq values.

Usage

```
cqDetectionMethodType(value = "")
```

Arguments

`value`	‘character(1)‘. Enumeration value; must be one of the allowed values listed below.

Format

An S7 class.

Value

An S7 object of class ‘cqDetectionMethodType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Allowed values

‘automated threshold and baseline settings‘ Allowed value.

‘manual threshold and baseline settings‘ Allowed value.

‘second derivative maximum‘ Allowed value.

‘other‘ Allowed value.

See Also

‘runType‘, ‘dataType‘

`dataCollectionSoftwareType` *dataCollectionSoftwareType S7 class*

Description

Software used by the instrument to collect qPCR data. Inherits from ‘rdmlBaseType‘.

Usage

```
dataCollectionSoftwareType(name = character(0), version = character(0))
```

Arguments

`name`	‘character(1)‘. Software name.
`version`	‘character(1)‘. Software version.

Format

An S7 class.

Value

An S7 object of class ‘dataCollectionSoftwareType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘name‘ ‘character(1)‘. Software name.

‘version‘ ‘character(1)‘. Software version.

See Also

‘runType‘

---

dataType *dataType S7 class*

Description

Target-specific qPCR result data for one reaction. It stores calculated results and optional amplification and melting fluorescence curves. Inherits from ‘rdmlBaseType‘.

Usage

```
dataType(
  targetId = NA,
  cq = NA_real_,
  N0 = NA_real_,
  ampEffMet = NA_character_,
  ampEff = NA_real_,
  ampEffSE = NA_real_,
  corrF = NA_real_,
  corrP = NA_real_,
  meltTemp = NA_real_,
  excl = NA_character_,
  note = NA_character_,
  adp = NA,
  mdp = NA,
  endPt = NA_real_,
  bgFluor = NA_real_,
  bgFluorSlp = NA_real_,
  quantFluor = NA_real_
)
```

Arguments

`targetId`	‘idReferenceType‘. Reference to the target.
`cq`	‘numeric(1)‘ or ‘NA‘. Quantification cycle (Cq).
`N0`	‘numeric(1)‘ or ‘NA‘. Estimated initial template quantity.
`ampEffMet`	‘character(1)‘ or ‘NA‘. Amplification-efficiency method.
`ampEff`	‘numeric(1)‘ or ‘NA‘. Amplification efficiency.
`ampEffSE`	‘numeric(1)‘ or ‘NA‘. Standard error of amplification efficiency.
`corrF`	‘numeric(1)‘ or ‘NA‘. Fluorescence correction factor.
`corrP`	‘numeric(1)‘ or ‘NA‘. Correction parameter.
`meltTemp`	Numeric vector or ‘NA‘. One or more melting temperatures.
`excl`	‘character(1)‘ or ‘NA‘. Exclusion information.
`note`	‘character(1)‘ or ‘NA‘. Free-text note.
`adp`	‘dpAmpCurveType‘ or ‘NA‘. Amplification curve.
`mdp`	‘dpMeltingCurveType‘ or ‘NA‘. Melting curve.
`endPt`	‘numeric(1)‘ or ‘NA‘. End-point fluorescence/result.
`bgFluor`	‘numeric(1)‘ or ‘NA‘. Background fluorescence.
`bgFluorSlp`	‘numeric(1)‘ or ‘NA‘. Background-fluorescence slope.
`quantFluor`	‘numeric(1)‘ or ‘NA‘. Quantification fluorescence.

Format

An S7 class.

Value

An S7 object of class ‘dataType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘targetId‘ ‘idReferenceType‘. Reference to the target.

‘cq‘ ‘numeric(1)‘ or ‘NA‘. Quantification cycle (Cq).

‘N0‘ ‘numeric(1)‘ or ‘NA‘. Estimated initial template quantity.

‘ampEffMet‘ ‘character(1)‘ or ‘NA‘. Amplification-efficiency method.

‘ampEff‘ ‘numeric(1)‘ or ‘NA‘. Amplification efficiency.

‘ampEffSE‘ ‘numeric(1)‘ or ‘NA‘. Standard error of amplification efficiency.

‘corrF‘ ‘numeric(1)‘ or ‘NA‘. Fluorescence correction factor.

‘corrP‘ ‘numeric(1)‘ or ‘NA‘. Correction parameter.

‘meltTemp‘ Numeric vector or ‘NA‘. One or more melting temperatures.

‘excl‘ ‘character(1)‘ or ‘NA‘. Exclusion information.

‘note‘ ‘character(1)‘ or ‘NA‘. Free-text note.

‘adp‘ ‘dpAmpCurveType‘ or ‘NA‘. Amplification curve.

‘mdp‘ ‘dpMeltingCurveType‘ or ‘NA‘. Melting curve.

‘endPt‘ ‘numeric(1)‘ or ‘NA‘. End-point fluorescence/result.

‘bgFluor‘ ‘numeric(1)‘ or ‘NA‘. Background fluorescence.

‘bgFluorSlp‘ ‘numeric(1)‘ or ‘NA‘. Background-fluorescence slope.

‘quantFluor‘ ‘numeric(1)‘ or ‘NA‘. Quantification fluorescence.

See Also

‘reactType‘, ‘targetType‘, ‘dpAmpCurveType‘, ‘dpMeltingCurveType‘

`detectRDMLFormat` *Detect the format handler selected for a path*

Description

Detect the format handler selected for a path

Usage

```
detectRDMLFormat(fileName, operation = c("read", "write"))
```

Arguments

`fileName`	File path.
`operation`	‘"read"‘ or ‘"write"‘.

Value

Registered format name.

See Also

‘listRDMLFormats‘, ‘readRDML‘, ‘writeRDML‘

Examples

```
amplificationFile <- system.file(
  "extdata", "lc96_bACTXY.rdml", package = "RDML7"
)
meltingFile <- system.file(
  "extdata", "BioRad_qPCR_melt.rdml", package = "RDML7"
)
detectRDMLFormat(amplificationFile, operation = "read")
detectRDMLFormat(meltingFile, operation = "write")
```

documentationType *documentationType S7 class*

Description

Reusable documentation text. Use this type when the same description or note is referenced by several samples, targets, runs, or experiments. Inherits from ‘rdmlBaseType‘.

Usage

```
documentationType(id = NULL, text = NA_character_)
```

Arguments

id	‘idType‘. Unique documentation identifier.
text	‘character(1)‘ or ‘NA‘. Documentation text.

Format

An S7 class.

Value

An S7 object of class ‘documentationType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘id‘ ‘idType‘. Unique documentation identifier.

‘text‘ ‘character(1)‘ or ‘NA‘. Documentation text.

See Also

‘rdmlType‘, ‘idReferenceType‘

dpAmpCurveType *dpAmpCurveType S7 class*

Description

Amplification fluorescence curve. Inherits from ‘rdmlBaseType‘.

Usage

```
dpAmpCurveType(fpoints = NULL)
```

Arguments

`fpoints` ‘data.table‘ containing ‘cyc‘ and ‘fluor‘, with optional per-cycle ‘tmp‘.

Format

An S7 class.

Value

An S7 object of class ‘dpAmpCurveType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘fpoints‘ ‘data.table‘ containing ‘cyc‘ and ‘fluor‘, with optional per-cycle ‘tmp‘.

See Also

‘dataType‘, ‘getFData‘, ‘setFData‘

`dpMeltingCurveType` *dpMeltingCurveType S7 class*

Description

Melting fluorescence curve. Inherits from ‘rdmlBaseType‘.

Usage

```
dpMeltingCurveType(fpoints = NULL)
```

Arguments

`fpoints` ‘data.table‘ containing ‘tmp‘ and ‘fluor‘.

Format

An S7 class.

Value

An S7 object of class ‘dpMeltingCurveType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘fpoints‘ ‘data.table‘ containing ‘tmp‘ and ‘fluor‘.

See Also

‘dataType‘, ‘getFData‘, ‘setFData‘

---

dyeChemistryType *dyeChemistryType S7 enumeration*

Description

Fluorescence chemistry used by a dye or detection system.

Usage

```
dyeChemistryType(value = "")
```

Arguments

`value`	‘character(1)‘. Enumeration value; must be one of the allowed values listed below.

Format

An S7 class.

Value

An S7 object of class ‘dyeChemistryType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Allowed values

‘non-saturating DNA binding dye‘ Allowed value.

‘saturating DNA binding dye‘ Allowed value.

‘hybridization probe‘ Allowed value.

‘hydrolysis probe‘ Allowed value.

‘labelled forward primer‘ Allowed value.

‘labelled reverse primer‘ Allowed value.

‘DNA-zyme probe‘ Allowed value.

See Also

‘dyeType‘

`dyeType` *dyeType S7 class*

Description

Description of a fluorescent dye or reporter used to detect a target. Inherits from ‘rdmlBaseType‘.

Usage

```
dyeType(id = NULL, description = NA_character_, dyeChemistry = NA)
```

Arguments

`id`	‘idType‘. Unique dye identifier.
`description`	‘character(1)‘ or ‘NA‘. Human-readable dye description.
`dyeChemistry`	‘dyeChemistryType‘ or ‘NA‘. Detection chemistry.

Format

An S7 class.

Value

An S7 object of class ‘dyeType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘id‘ ‘idType‘. Unique dye identifier.

‘description‘ ‘character(1)‘ or ‘NA‘. Human-readable dye description.

‘dyeChemistry‘ ‘dyeChemistryType‘ or ‘NA‘. Detection chemistry.

See Also

‘dyeChemistryType‘, ‘targetType‘

editRDML — *Launch the RDML editing interface*

Description

Starts the optional Shiny-based editor when its GUI dependencies are installed.

Usage

```
editRDML()
```

Value

Result of launching the application, invisibly where applicable.

See Also

‘readRDML‘, ‘writeRDML‘

experimenterType — *experimenterType S7 class*

Description

Contact details of an experimenter associated with an RDML document or thermal cycling conditions. Inherits from ‘rdmlBaseType‘.

Usage

```
experimenterType(
  id = NULL,
  firstName = character(0),
  lastName = character(0),
  email = NA_character_,
  labName = NA_character_,
  labAddress = NA_character_
)
```

Arguments

id	‘idType‘. Unique experimenter identifier.
firstName	‘character(1)‘. First name.
lastName	‘character(1)‘. Last name.
email	‘character(1)‘ or ‘NA‘. E-mail address.
labName	‘character(1)‘ or ‘NA‘. Laboratory or organization name.
labAddress	‘character(1)‘ or ‘NA‘. Laboratory postal address.

Format

An S7 class.

Value

An S7 object of class ‘experimenterType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘id‘ ‘idType‘. Unique experimenter identifier.

‘firstName‘ ‘character(1)‘. First name.

‘lastName‘ ‘character(1)‘. Last name.

‘email‘ ‘character(1)‘ or ‘NA‘. E-mail address.

‘labName‘ ‘character(1)‘ or ‘NA‘. Laboratory or organization name.

‘labAddress‘ ‘character(1)‘ or ‘NA‘. Laboratory postal address.

See Also

‘rdmlType‘, ‘thermalCyclingConditionsType‘

`experimentType` *experimentType S7 class*

Description

A qPCR experiment. One experiment may contain several instrument runs. Inherits from ‘rdmlBaseType‘.

Usage

```
experimentType(
  id = NULL,
  description = NA_character_,
  documentation = NA,
  run = NA
)
```

Arguments

`id`	‘idType‘. Unique experiment identifier.
`description`	‘character(1)‘ or ‘NA‘. Experiment description.
`documentation`	List of ‘idReferenceType‘ objects or ‘NA‘. Documentation references.
`run`	Run-id-keyed list of ‘runType‘ objects or ‘NA‘.

Format

An S7 class.

Value

An S7 object of class ‘experimentType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘id‘ ‘idType‘. Unique experiment identifier.

‘description‘ ‘character(1)‘ or ‘NA‘. Experiment description.

‘documentation‘ List of ‘idReferenceType‘ objects or ‘NA‘. Documentation references.

‘run‘ Run-id-keyed list of ‘runType‘ objects or ‘NA‘.

See Also

‘runType‘, ‘rdmlType‘

`getFData` *Extract fluorescence data*

Description

Extracts amplification or melting curves either from one [dataType] object or from a selection of curves in an [rdmlType] object.

Usage

```
getFData(x, ...)
```

Arguments

`x`	A [dataType] containing one target-specific result, or an [rdmlType] containing complete experiments.
`...`	Reserved for method-specific extensions.

Value

A [data.table::data.table]. See **Output columns** for the shape returned by each method and argument combination.

Method arguments

‘dpType‘ Character scalar accepted by both methods: ‘"adp"‘ for amplification data or ‘"mdp"‘ for melting data. Defaults to ‘"adp"‘.

‘request‘ For an ‘rdmlType‘, a data frame normally produced by [asTable()]. It must contain ‘fdataName‘, ‘expId‘, ‘runId‘, ‘reactId‘, and ‘target‘. Additional columns are retained in long output. If omitted, ‘asTable(x)‘ selects every stored target result.

‘longTable‘ For an ‘rdmlType‘, return one row per measured point with request metadata repeated (‘TRUE‘), or a wide table with one fluorescence column per ‘fdataName‘ (‘FALSE‘, the default).

‘includeMissing‘ In long form, retain requested metadata rows that have no curve of the selected ‘dpType‘, filling point columns with ‘NA‘? Defaults to ‘FALSE‘. It has no effect on wide output.

Output columns

For a single ‘dataType‘, amplification output has ‘cyc‘, optional ‘tmp‘, and ‘fluor‘; melting output has ‘tmp‘ and ‘fluor‘. A missing curve returns an empty table with the appropriate coordinate and ‘fluor‘ columns.

For an ‘rdmlType‘, wide amplification output begins with ‘cyc‘ and may include ‘tmp‘ when temperature is unambiguous for every cycle. Wide melting output begins with ‘tmp‘. Remaining columns are named by ‘fdataName‘ in the order requested. Long output joins the request columns to ‘cyc‘/‘tmp‘ and ‘fluor‘.

Duplicate ‘fdataName‘ values are replaced by unique names constructed from ‘expId‘, ‘runId‘, ‘reactId‘, and ‘target‘, with a warning.

See Also

‘asTable()‘, ‘setFData()‘

Examples

```
rdmlFile <- system.file("extdata", "lc96_bACTXY.rdml", package = "RDML7")
rdml <- readRDML(rdmlFile, showProgress = FALSE)
meta <- asTable(
  rdml,
  columns = c(
    "expId", "runId", "reactId", "position", "sample", "target", "adp"
  )
)

# All amplification curves as one wide table.
amplification <- getFData(rdml, dpType = "adp")

# Selected curves with metadata repeated for every cycle.
selected <- getFData(
  rdml,
  request = meta[seq_len(min(2L, nrow(meta))), ],
  dpType = "adp",
```

```
  longTable = TRUE
)

# Keep metadata even when melting data are absent.
melting <- getFData(
  rdml,
  request = meta,
  dpType = "mdp",
  longTable = TRUE,
  includeMissing = TRUE
)
```

`gradientType` *gradientType S7 class*

Description

Temperature-gradient step. Inherits from ‘temperatureBaseType‘.

Usage

```
gradientType(
  duration = integer(0),
  temperatureChange = NA_real_,
  durationChange = NA_integer_,
  measure = NA,
  ramp = NA_real_,
  highTemperature = integer(0),
  lowTemperature = integer(0)
)
```

Arguments

`duration`	Positive integer. Step duration.
`temperatureChange`	‘numeric(1)‘ or ‘NA‘. Temperature increment/decrement per cycle.
`durationChange`	‘integer(1)‘ or ‘NA‘. Duration increment/decrement per cycle.
`measure`	‘measureType‘ or ‘NA‘. Fluorescence measurement mode.
`ramp`	‘numeric(1)‘ or ‘NA‘. Ramp rate.
`highTemperature`	‘numeric(1)‘. High end of the gradient.
`lowTemperature`	‘numeric(1)‘. Low end of the gradient.

Format

An S7 class.

Value

An S7 object of class ‘gradientType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘highTemperature‘ ‘numeric(1)‘. High end of the gradient.

‘lowTemperature‘ ‘numeric(1)‘. Low end of the gradient.

See Also

‘temperatureBaseType‘, ‘stepType‘

`idReferenceType` *RDML identifier reference*

Description

Extends ‘idType‘ for references between RDML elements, for example sample, target, dye, documentation, and thermal-program references.

Usage

```
idReferenceType(id = character(0))
```

Arguments

`id`	Character identifier of the referenced RDML element.

Value

An S7 object of class ‘idReferenceType‘ representing a reference from one RDML schema element to another identified element.

See Also

‘idType‘, ‘validateRDML‘

idType — *RDML identifier*

Description

Stores one character identifier used as a schema key. ‘as.character()‘ extracts the identifier and ‘print()‘ displays it directly.

Usage

```
idType(id = character(0))
```

Arguments

id	Character identifier.

Value

An S7 object of class ‘idType‘ representing an identifier used as a key in the RDML object hierarchy.

See Also

‘idReferenceType‘, ‘rdmlKeyedList‘, ‘rdmlType‘

isValidRDML — *Test whether an RDML object has semantic errors*

Description

Test whether an RDML object has semantic errors

Usage

```
isValidRDML(x, level = "full")
```

Arguments

x	‘rdmlType‘.
level	Validation level passed to ‘validateRDML()‘.

Value

‘TRUE‘ if no validation issue has severity ‘"error"‘.

See Also

‘validateRDML‘

Examples

```
rdmlFile <- system.file(
  "extdata", "BioRad_qPCR_melt.rdml", package = "RDML7"
)
rdml <- readRDML(rdmlFile, showProgress = FALSE)

isValidRDML(rdml, level = "full")
```

---

labelFormatType *labelFormatType S7 enumeration*

Description

Labeling convention used for plate rows or columns.

Usage

```
labelFormatType(value = "")
```

Arguments

`value`	‘character(1)‘. Enumeration value; must be one of the allowed values listed below.

Format

An S7 class.

Value

An S7 object of class ‘labelFormatType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Allowed values

‘ABC‘ Allowed value.

‘123‘ Allowed value.

‘A1a1‘ Allowed value.

See Also

‘pcrFormatType‘

legacy-api *Legacy API compatibility wrappers*

Description

These functions preserve public names, dotted argument names, and (where applicable) dotted output-column names used by previous RDML releases. New code should use the canonical camelCase API.

Usage

```
AsTable(
  x,
  .default = list(exp.id = .rdmlIdChr(experiment$id), run.id = .rdmlIdChr(run$id),
   react.id = .rdmlIdChr(react$id), position = .rdmlReactPosition(react, run$pcrFormat),
   sample = .rdmlIdChr(react$sample), target = .rdmlIdChr(data$targetId), target.dyeId =
     .rdmlTargetDye(target, .rdmlIdChr(data$targetId)), sample.type =
   .rdmlSampleType(sample[[.rdmlIdChr(react$sample)]], .rdmlIdChr(data$targetId)), adp =
     .rdmlPresent(data$adp), mdp = .rdmlPresent(data$mdp)),
  name.pattern = paste(.rdmlIdChr(experiment$id), .rdmlIdChr(run$id),
     .rdmlReactPosition(react, run$pcrFormat), .rdmlIdChr(react$sample),
   .rdmlSampleType(sample[[.rdmlIdChr(react$sample)]], .rdmlIdChr(data$targetId)),
     .rdmlIdChr(data$targetId), sep = "_"),
  add.columns = list(),
  treat.null.as.na = FALSE,
  include.hidden = FALSE,
  ...
)

GetFData(x, request, dp.type = "adp", long.table = FALSE, ...)

SetFData(
  x,
  fdata,
  description,
  fdata.type = "adp",
  conflict = c("error", "keep", "replace"),
  ...
)

AsDendrogram(x, plot.dendrogram = TRUE, ...)

MergeRDMLs(to.merge, data.conflict = c("incoming", "base", "error"))

rdmlEdit(...)
```

Arguments

`x`	An [rdmlType] object, or a [dataType] for ‘GetFData()‘.
`.default`	Named list of expressions defining the base ‘AsTable()‘ output columns. See [asTable()] for values available during evaluation.
`name.pattern`	Expression defining one legacy ‘fdata.name‘ per target result.
`add.columns`	Named list of additional ‘AsTable()‘ expressions.
`treat.null.as.na`	Logical. Convert ‘NULL‘ expression results to ‘NA‘.
`include.hidden`	Logical. Include experiment identifiers beginning with ‘.‘.
`...`	Additional arguments passed unchanged to the canonical API.
`request`	Legacy metadata table describing curves to extract. Required columns are ‘fdata.name‘, ‘exp.id‘, ‘run.id‘, ‘react.id‘, and ‘target‘.
`dp.type`	Curve type: ‘"adp"‘ or ‘"mdp"‘.
`long.table`	Logical. Return long rather than wide fluorescence data.
`fdata`	Wide fluorescence matrix or table; the first column is ‘cyc‘ for ‘"adp"‘ or ‘tmp‘ for ‘"mdp"‘.
`description`	Legacy metadata table describing ‘fdata‘ columns.
`fdata.type`	Curve type: ‘"adp"‘ or ‘"mdp"‘.
`conflict`	Metadata conflict policy: ‘"error"‘, ‘"keep"‘, or ‘"replace"‘.
`plot.dendrogram`	Logical. Plot the generated dendrogram.
`to.merge`	Non-empty list of [rdmlType] objects.
`data.conflict`	Duplicate target-data policy: ‘"incoming"‘, ‘"base"‘, or ‘"error"‘.

Details

Legacy	Canonical
‘AsTable()‘	‘asTable()‘
‘GetFData()‘	‘getFData()‘
‘SetFData()‘	‘setFData()‘
‘AsDendrogram()‘	‘asDendrogram()‘
‘MergeRDMLs()‘	‘mergeRDMLs()‘
‘rdmlEdit()‘	‘editRDML()‘

Legacy ‘AsTable()‘ returns names such as ‘fdata.name‘, ‘exp.id‘, ‘run.id‘, ‘react.id‘, ‘sample.type‘, and ‘target.dyeId‘; ‘asTable()‘ uses ‘fdataName‘, ‘expId‘, ‘runId‘, ‘reactId‘, ‘sampleType‘, and ‘targetDyeId‘.

Value

‘AsTable()‘ and ‘GetFData()‘ return [data.table::data.table] objects with legacy dotted column names where applicable. ‘SetFData()‘ and ‘MergeRDMLs()‘ return [rdmlType] objects. ‘AsDendrogram()‘ returns a ‘dendrogram‘. ‘rdmlEdit()‘ launches the editor like [editRDML()].

See Also

[asTable()], [getFData()], [setFData()], [asDendrogram()], [mergeRDMLs()], [editRDML()], ‘RDML7-package‘

Examples

```
amplificationFile <- system.file(
  "extdata", "lc96_bACTXY.rdml", package = "RDML7"
)
meltingFile <- system.file(
  "extdata", "BioRad_qPCR_melt.rdml", package = "RDML7"
)
amplification <- readRDML(amplificationFile, showProgress = FALSE)
melting <- readRDML(meltingFile, showProgress = FALSE)

legacyMeta <- AsTable(amplification)
legacyCurves <- GetFData(
  amplification,
  request = legacyMeta[seq_len(min(2L, nrow(legacyMeta))), ],
  dp.type = "adp"
)
copied <- SetFData(
  rdmlType(),
  legacyCurves,
  legacyMeta[seq_len(min(2L, nrow(legacyMeta))), ],
  fdata.type = "adp"
)
tree <- AsDendrogram(melting, plot.dendrogram = FALSE)
merged <- MergeRDMLs(list(amplification, melting), data.conflict = "base")
```

---

lidOpenType *lidOpenType S7 class*

Description

Marker step indicating that the instrument lid is opened. This class has no additional properties. Inherits from ‘rdmlBaseType‘.

Usage

```
lidOpenType()
```

Format

An S7 class.

Value

An S7 object of class ‘lidOpenType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

This class does not define additional properties.

See Also

‘stepType‘

---

listRDMLFormats *List registered RDML file formats*

Description

Formats are returned in registration order. For automatic dispatch, the first format supporting a given extension and operation is the default.

Usage

```
listRDMLFormats()
```

Value

Data frame with format name, extensions, and reader/writer availability.

See Also

‘registerRDMLFormat‘, ‘detectRDMLFormat‘

---

loadRDMLModule *Load file-format handlers from an R module*

Description

The module defines ‘rdmlModule()‘ (legacy ‘rdml_module()‘ is also accepted) and returns one or more specifications accepted by ‘registerRDMLFormat()‘. Each specification may contain only ‘name‘, ‘extensions‘, ‘reader‘, and ‘writer‘.

Usage

```
loadRDMLModule(path)
```

Arguments

path	Module R file.

Value

Registered format names invisibly.

See Also

‘registerRDMLFormat‘, ‘listRDMLFormats‘

---

loopType *loopType S7 class*

Description

Loop instruction in a thermal-cycling program. Inherits from ‘rdmlBaseType‘.

Usage

```
loopType(goto = integer(0), `repeat` = integer(0))
```

Arguments

`goto`	Positive integer. Number of the step to return to.
`repeat`	Positive integer. Number of loop repetitions.

Format

An S7 class.

Value

An S7 object of class ‘loopType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘goto‘ Positive integer. Number of the step to return to.

‘repeat‘ Positive integer. Number of loop repetitions.

See Also

‘stepType‘, ‘thermalCyclingConditionsType‘

measureType — *measureType S7 enumeration*

Description

Fluorescence measurement mode used in a thermal-cycling step.

Usage

```
measureType(value = "")
```

Arguments

`value`	‘character(1)‘. Enumeration value; must be one of the allowed values listed below.

Format

An S7 class.

Value

An S7 object of class ‘measureType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Allowed values

‘real time‘ Allowed value.

‘meltcurve‘ Allowed value.

See Also

‘temperatureBaseType‘

mergeRDMLs — *Merge multiple RDML objects*

Description

Merge two or more [rdmlType] objects recursively.

Usage

```
mergeRDMLs(..., dataConflict = c("incoming", "base", "error"))
```

Arguments

...	Two or more ‘rdmlType‘ objects, or a single non-empty list of ‘rdmlType‘ objects.
dataConflict	Conflict policy for duplicate target data: ‘"incoming"‘, ‘"base"‘, or ‘"error"‘.

Details

The **first** RDML object is used as the metadata base. Every subsequent object is merged into the accumulated result from left to right:

‘mergeRDMLs(rdml1, rdml2, rdml3)‘ is therefore equivalent to ‘merge(merge(rdml1, rdml2), rdml3)‘.

The function accepts either RDML objects as separate arguments:

‘mergeRDMLs(rdml1, rdml2, rdml3)‘

or a single list containing RDML objects:

‘mergeRDMLs(list(rdml1, rdml2, rdml3))‘.

Top-level keyed metadata (‘experimenter‘, ‘documentation‘, ‘dye‘, ‘sample‘, ‘target‘, and ‘thermalCyclingConditions‘) keep the definition from the accumulated base object when the same key is present in both objects.

‘dataConflict‘ applies to duplicate target data within the same reaction:

- ‘"incoming"‘: replace base data with data from the current incoming object. With multiple objects, later objects therefore have higher priority. - ‘"base"‘: preserve data already present in the accumulated base object. With multiple objects, earlier objects therefore have higher priority. - ‘"error"‘: stop at the first duplicate data target.

Value

Merged ‘rdmlType‘.

See Also

[readRDML()], [writeRDML()]

Examples

```
amplificationFile <- system.file(
  "extdata", "lc96_bACTXY.rdml", package = "RDML7"
)
meltingFile <- system.file(
  "extdata", "BioRad_qPCR_melt.rdml", package = "RDML7"
)

amplification <- readRDML(amplificationFile, showProgress = FALSE)
melting <- readRDML(meltingFile, showProgress = FALSE)

# Separate arguments
merged <- mergeRDMLs(
  amplification,
  melting,
```

```
  dataConflict = "error"
)

# A list is also accepted
mergedFromList <- mergeRDMLs(
  list(amplification, melting),
  dataConflict = "base"
)
```

`metadata-schema-types` *Metadata schema types*

Description

S7 classes representing reusable metadata elements in an RDML document.

Details

‘experimenterType‘ Experimenter/contact metadata: ‘id‘, ‘firstName‘, ‘lastName‘, optional ‘email‘, ‘labName‘, and ‘labAddress‘.

‘documentationType‘ Reusable documentation identified by ‘id‘ and optional free text.

‘dyeChemistryType‘ Enumeration describing fluorescence chemistry.

‘dyeType‘ Dye metadata: ‘id‘, optional ‘description‘, and optional ‘dyeChemistry‘.

‘xRefType‘ External cross-reference with optional ‘name‘ and ‘id‘.

‘annotationType‘ Annotation ‘property‘ / ‘value‘ pair.

‘rdmlIdType‘ Top-level publisher identifier containing ‘publisher‘, ‘serialNumber‘, and optional ‘MD5Hash‘.

See Also

‘idType‘, ‘idReferenceType‘, ‘rdmlType‘

`newRDMLImportData` *Create canonical importer data*

Description

Create canonical importer data

Usage

```
newRDMLImportData(
  series,
  publisher = NA_character_,
  serialNumber = "1",
  format = NA_character_,
  preserveReactIds = FALSE,
  metadata = list(),
  losses = list()
)
```

Arguments

series	List of ‘rdmlImportSeries‘ objects.
publisher	Optional source/device publisher.
serialNumber	Source/device serial identifier.
format	Source-format identifier.
preserveReactIds	Preserve supplied reaction ids literally.
metadata	Additional importer metadata.
losses	List of loss records.

Value

‘rdmlImportData‘.

See Also

[newRDMLImportSeries()], [buildRDMLImport()]

Examples

```
rdmlFile <- system.file(
  "extdata", "BioRad_qPCR_melt.rdml", package = "RDML7"
)
rdml <- readRDML(rdmlFile, showProgress = FALSE)
description <- asTable(rdml)
description <- description[description$mdp %in% TRUE, ]
description <- description[seq_len(min(2L, nrow(description))), ]
fluorescence <- getFData(rdml, request = description, dpType = "mdp")
series <- newRDMLImportSeries("mdp", fluorescence, description)

importData <- newRDMLImportData(
  series = list(series),
  publisher = "Bio-Rad",
  serialNumber = "example",
  format = "rdml",
  metadata = list(source = basename(rdmlFile))
)
```

```
names(importData)
```

`newRDMLImportSeries` *Create an importer fluorescence-series object*

Description

Create an importer fluorescence-series object

Usage

```
newRDMLImportSeries(fdataType, fdata, description)
```

Arguments

`fdataType`	‘"adp"‘ or ‘"mdp"‘.
`fdata`	Wide fluorescence table accepted by ‘setFData()‘.
`description`	Description table accepted by ‘setFData()‘.

Value

‘rdmlImportSeries‘.

See Also

[newRDMLImportData()], [buildRDMLImport()]

Examples

```
rdmlFile <- system.file("extdata", "lc96_bACTXY.rdml", package = "RDML7")
rdml <- readRDML(rdmlFile, showProgress = FALSE)
description <- asTable(rdml)
description <- description[seq_len(min(2L, nrow(description))), ]
fluorescence <- getFData(rdml, request = description, dpType = "adp")

series <- newRDMLImportSeries(
  fdataType = "adp",
  fdata = fluorescence,
  description = description
)
series$fdataType
```

newRDMLLossRecord *Create a structured lossy-conversion record*

Description

Importers/exporters use these records for information that cannot be represented exactly in the destination model.

Usage

```
newRDMLLossRecord(code, message, path = NA_character_, details = list())
```

Arguments

code	Machine-readable loss code.
message	Human-readable description.
path	Optional RDML object path.
details	Additional structured metadata.

Value

An ‘newRDMLLossRecord‘.

See Also

‘readRDML‘, ‘writeRDML‘, ‘rdmlImportData‘

Examples

```
rdmlFile <- system.file(
  "extdata", "BioRad_qPCR_melt.rdml", package = "RDML7"
)
loss <- newRDMLLossRecord(
  code = "unsupportedMetadata",
  message = "Vendor metadata is retained only in the importer record",
  path = rdmlFile,
  details = list(format = detectRDMLFormat(rdmlFile, "read"))
)
loss
```

`nucleotideType` *nucleotideType S7 enumeration*

Description

Nucleic-acid type of the template material.

Usage

```
nucleotideType(value = "")
```

Arguments

`value`	‘character(1)‘. Enumeration value; must be one of the allowed values listed below.

Format

An S7 class.

Value

An S7 object of class ‘nucleotideType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Allowed values

‘DNA‘ Allowed value.

‘genomic DNA‘ Allowed value.

‘cDNA‘ Allowed value.

‘RNA‘ Allowed value.

See Also

‘templateQuantityType‘

`oligoType` *oligoType S7 class*

Description

Oligonucleotide sequence used for a primer, probe, or amplicon definition. Inherits from ‘rdmlBaseType‘.

Usage

```
oligoType(
  threePrimeTag = NA_character_,
  fivePrimeTag = NA_character_,
  sequence = character(0)
)
```

Arguments

`threePrimeTag`	‘character(1)‘ or ‘NA‘. Optional 3-prime modification/tag.
`fivePrimeTag`	‘character(1)‘ or ‘NA‘. Optional 5-prime modification/tag.
`sequence`	‘character(1)‘. Nucleotide sequence.

Format

An S7 class.

Value

An S7 object of class ‘oligoType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘threePrimeTag‘ ‘character(1)‘ or ‘NA‘. Optional 3-prime modification/tag.

‘fivePrimeTag‘ ‘character(1)‘ or ‘NA‘. Optional 5-prime modification/tag.

‘sequence‘ ‘character(1)‘. Nucleotide sequence.

See Also

‘sequencesType‘

partitionDataType *partitionDataType S7 class*

Description

Target-specific digital-PCR partition result data. Inherits from ‘rdmlBaseType‘.

Usage

```
partitionDataType(
  targetId = NA,
  excluded = NA_character_,
  note = NA_character_,
  pos = integer(0),
  neg = integer(0),
  undef = NA_integer_,
  excl = NA_integer_,
  conc = NA_real_
)
```

Arguments

`targetId`	‘idReferenceType‘. Reference to the target.
`excluded`	‘character(1)‘ or ‘NA‘. Exclusion information.
`note`	‘character(1)‘ or ‘NA‘. Note.
`pos`	Positive integer. Positive partitions.
`neg`	Positive integer. Negative partitions.
`undef`	Positive integer or ‘NA‘. Undefined partitions.
`excl`	Positive integer or ‘NA‘. Excluded partitions.
`conc`	‘numeric(1)‘ or ‘NA‘. Calculated concentration.

Format

An S7 class.

Value

An S7 object of class ‘partitionDataType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘targetId‘ ‘idReferenceType‘. Reference to the target.

‘excluded‘ ‘character(1)‘ or ‘NA‘. Exclusion information.

‘note‘ ‘character(1)‘ or ‘NA‘. Note.

‘pos‘ Positive integer. Positive partitions.

‘neg‘ Positive integer. Negative partitions.

‘undef‘ Positive integer or ‘NA‘. Undefined partitions.

‘excl‘ Positive integer or ‘NA‘. Excluded partitions.

‘conc‘ ‘numeric(1)‘ or ‘NA‘. Calculated concentration.

See Also

‘partitionsType‘, ‘targetType‘

`partitionsType` *partitionsType S7 class*

Description

Digital-PCR partition metadata and target-specific partition results. Inherits from ‘rdmlBaseType‘.

Usage

```
partitionsType(volume = integer(0), endPtTable = NA_character_, data = NA)
```

Arguments

`volume`	‘numeric(1)‘. Partition volume.
`endPtTable`	‘character(1)‘ or ‘NA‘. Optional endpoint-table representation/reference.
`data`	Target-keyed list of ‘partitionDataType‘ objects or ‘NA‘.

Format

An S7 class.

Value

An S7 object of class ‘partitionsType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘volume‘ ‘numeric(1)‘. Partition volume.

‘endPtTable‘ ‘character(1)‘ or ‘NA‘. Optional endpoint-table representation/reference.

‘data‘ Target-keyed list of ‘partitionDataType‘ objects or ‘NA‘.

See Also

‘partitionDataType‘, ‘reactType‘

`pauseType` *pauseType S7 class*

Description

Pause instruction at a specified temperature. Inherits from ‘rdmlBaseType‘.

Usage

```
pauseType(temperature = integer(0))
```

Arguments

`temperature` ‘numeric(1)‘. Pause temperature.

Format

An S7 class.

Value

An S7 object of class ‘pauseType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘temperature‘ ‘numeric(1)‘. Pause temperature.

See Also

‘stepType‘

`pcrFormatType` *pcrFormatType S7 class*

Description

PCR plate or reaction-layout dimensions and labeling conventions. Inherits from ‘rdmlBaseType‘.

Usage

```
pcrFormatType(
  rows = integer(0),
  columns = integer(0),
  rowLabel = NA,
  columnLabel = NA
)
```

Arguments

`rows`	Positive integer. Number of rows.
`columns`	Positive integer. Number of columns.
`rowLabel`	‘labelFormatType‘. Row-label convention.
`columnLabel`	‘labelFormatType‘. Column-label convention.

Format

An S7 class.

Value

An S7 object of class ‘pcrFormatType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘rows‘ Positive integer. Number of rows.

‘columns‘ Positive integer. Number of columns.

‘rowLabel‘ ‘labelFormatType‘. Row-label convention.

‘columnLabel‘ ‘labelFormatType‘. Column-label convention.

See Also

‘labelFormatType‘, ‘runType‘

`primingMethodType` *primingMethodType S7 enumeration*

Description

Priming strategy used for cDNA synthesis.

Usage

```
primingMethodType(value = "")
```

Arguments

`value`	‘character(1)‘. Enumeration value; must be one of the allowed values listed below.

Format

An S7 class.

Value

An S7 object of class ‘primingMethodType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Allowed values

‘oligo-dt‘ Allowed value.

‘random‘ Allowed value.

‘target-specific‘ Allowed value.

‘oligo-dt and random‘ Allowed value.

‘other‘ Allowed value.

See Also

‘cdnaSynthesisMethodType‘

`quantityType` *quantityType S7 class*

Description

Target-specific quantity assigned to a sample. Inherits from ‘rdmlBaseType‘.

Usage

```
quantityType(targetId = NA, value = integer(0), unit = quantityUnitType())
```

Arguments

`targetId`	‘idReferenceType‘. Reference to the target.
`value`	‘numeric(1)‘. Quantity value.
`unit`	‘quantityUnitType‘. Unit of the quantity.

Format

An S7 class.

Value

An S7 object of class ‘quantityType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘targetId‘ ‘idReferenceType‘. Reference to the target.

‘value‘ ‘numeric(1)‘. Quantity value.

‘unit‘ ‘quantityUnitType‘. Unit of the quantity.

See Also

‘sampleType‘, ‘quantityUnitType‘

`quantityUnitType` *quantityUnitType S7 enumeration*

Description

Unit used for a target-specific sample quantity.

Usage

```
quantityUnitType(value = "")
```

Arguments

`value`	‘character(1)‘. Enumeration value; must be one of the allowed values listed below.

Format

An S7 class.

Value

An S7 object of class ‘quantityUnitType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Allowed values

‘cop‘ copies.

‘fold‘ fold value.

‘dil‘ dilution.

‘ng‘ nanograms.

‘nMol‘ nanomoles.

‘other‘ another unit.

See Also

‘quantityType‘

rdml-experimental-types

Experimental data and run schema types

Description

S7 classes for fluorescence curves, reactions, runs, experiments, plate layouts, digital-PCR partitions, and the top-level RDML document.

Curve and result classes

‘dpAmpCurveType‘ Amplification ‘fpoints‘: ‘cyc‘, optional ‘tmp‘, and ‘fluor‘.

‘dpMeltingCurveType‘ Melting ‘fpoints‘: ‘tmp‘ and ‘fluor‘.

‘dataType‘ Target-specific results and curves.

Hierarchy and layout

‘partitionDataType‘, ‘partitionsType‘ Digital-PCR partition data.

‘reactType‘ Reaction with sample reference and target-keyed data.

‘dataCollectionSoftwareType‘ Software name/version.

‘labelFormatType‘ Plate label enumeration.

‘pcrFormatType‘ Plate dimensions and label formats.

‘cqDetectionMethodType‘ Cq-detection method enumeration.

‘runType‘ Instrument run and reactions.

‘experimentType‘ Experiment and runs.

‘rdmlType‘ Top-level RDML document.

rdml-sample-target-types

Sample and target schema types

Description

S7 classes used to describe samples, quantities, targets, oligonucleotides, assays, and cDNA synthesis metadata.

Enumerations

‘sampleTypeType‘ ‘unkn‘, ‘ntc‘, ‘nac‘, ‘std‘, ‘ntp‘, ‘nrt‘, ‘pos‘, or ‘opt‘.

‘quantityUnitType‘ ‘cop‘, ‘fold‘, ‘dil‘, ‘ng‘, ‘nMol‘, or ‘other‘.

‘primingMethodType‘ ‘oligo-dt‘, ‘random‘, ‘target-specific‘, ‘oligo-dt and random‘, or ‘other‘.

‘nucleotideType‘ ‘DNA‘, ‘genomic DNA‘, ‘cDNA‘, or ‘RNA‘.

‘targetTypeType‘ ‘ref‘ or ‘toi‘.

Classes

‘sampleTargetType‘ Associates ‘targetId‘ with a sample type.

‘quantityType‘ Target-specific quantity: ‘targetId‘, ‘value‘, and ‘unit‘.

‘cdnaSynthesisMethodType‘ Enzyme, priming method, DNase treatment, and optional thermal-program reference.

‘templateQuantityType‘ Template concentration and nucleotide type.

‘sampleType‘ Sample metadata, target-specific type/quantity, annotations, documentation, and cDNA synthesis information.

‘oligoType‘ Oligonucleotide sequence and optional terminal tags.

‘sequencesType‘ Primers, probes, and optional amplicon sequence.

‘commercialAssayType‘ Commercial assay company and order number.

‘targetType‘ Target type, dye reference, amplification efficiency, optional sequences, and commercial assay metadata.

---

rdml-thermal-types *Thermal-cycling schema types*

Description

S7 types used to represent PCR and melting thermal programs.

Details

‘measureType‘ accepts ‘real time‘ or ‘meltcurve‘.

‘temperatureBaseType‘ Common duration, measurement, ramp, and change fields.

‘temperatureType‘ Fixed-temperature step.

‘gradientType‘ Gradient step with high/low temperatures.

‘loopType‘ Loop control with ‘goto‘ and ‘repeat‘.

‘pauseType‘ Pause at a temperature.

‘lidOpenType‘ Marker step indicating an open lid.

‘stepType‘ Numbered step containing one thermal action.

‘thermalCyclingConditionsType‘ Named thermal program containing ‘stepType‘ objects.

`rdmlIdType` *rdmlIdType S7 class*

Description

Publisher identifier for an RDML document. Inherits from ‘rdmlBaseType‘.

Usage

```
rdmlIdType(
  publisher = character(0),
  serialNumber = character(0),
  MD5Hash = NA_character_
)
```

Arguments

`publisher`	‘character(1)‘. Publisher/producer identifier.
`serialNumber`	‘character(1)‘. Serial number assigned by the publisher.
`MD5Hash`	‘character(1)‘ or ‘NA‘. Optional MD5 hash.

Format

An S7 class.

Value

An S7 object of class ‘rdmlIdType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘publisher‘ ‘character(1)‘. Publisher/producer identifier.

‘serialNumber‘ ‘character(1)‘. Serial number assigned by the publisher.

‘MD5Hash‘ ‘character(1)‘ or ‘NA‘. Optional MD5 hash.

See Also

‘rdmlType‘

rdmlImportData *Parsed importer data before construction of rdmlType*

Description

Decouples vendor parsing from construction of the nested RDML hierarchy. ‘readRDML()‘ builds this representation centrally with ‘buildRDMLImport()‘.

Usage

```
rdmlImportData(
  series,
  publisher = NA_character_,
  serialNumber = "1",
  format = NA_character_,
  preserveReactIds = FALSE,
  metadata = list(),
  losses = list()
)
```

Arguments

series	List of ‘rdmlImportSeries‘ objects.
publisher	Optional source/device publisher.
serialNumber	Source/device serial identifier.
format	Source-format identifier.
preserveReactIds	Preserve supplied well/reaction ids literally.
metadata	Additional importer metadata.
losses	List of ‘newRDMLLossRecord()‘ objects.

Value

An S7 object of class ‘rdmlImportData‘ containing normalized amplification and melting series together with source metadata and loss records, ready for construction of an RDML document.

See Also

‘rdmlImportSeries‘, ‘buildRDMLImport‘, ‘registerRDMLFormat‘

rdmlImportSeries *One fluorescence-series group for the importer API*

Description

Intermediate representation used by table/vendor parsers.

Usage

```
rdmlImportSeries(fdataType = character(0), fdata = NULL, description = NULL)
```

Arguments

fdataType	‘"adp"‘ or ‘"mdp"‘.
fdata	Wide fluorescence table accepted by ‘setFData()‘.
description	CamelCase description table accepted by ‘setFData()‘.

Value

An S7 object of class ‘rdmlImportSeries‘ containing one normalized fluorescence series and its associated experiment, run, reaction, sample, and target metadata.

See Also

‘rdmlImportData‘, ‘buildRDMLImport‘

rdmlKeyedList *Keyed list view used by RDML S7 objects*

Description

Derives virtual element names from a property such as ‘id‘ or ‘targetId‘ and enables nested ‘$‘ / ‘[[‘ access without storing physical list names.

Usage

```
rdmlKeyedList(x = list(), key = "id")
```

Arguments

x	List of schema objects.
key	Property used as the virtual key.

Value

An ‘rdmlKeyedList‘ S7 object.

See Also

‘rdmlType‘, ‘idType‘

---

rdmlType *rdmlType S7 class*

Description

Top-level container representing one RDML document. Its structure mirrors the RDML schema and stores reusable metadata together with experiments and runs. Inherits from ‘rdmlBaseType‘.

Usage

```
rdmlType(
  dateMade = NA_character_,
  dateUpdated = NA_character_,
  id = NA,
  experimenter = NA,
  documentation = NA,
  dye = NA,
  sample = NA,
  target = NA,
  thermalCyclingConditions = NA,
  experiment = NA
)
```

Arguments

dateMade	Date-time value or ‘NA‘. Document creation date.
dateUpdated	Date-time value or ‘NA‘. Last update date.
id	List of ‘rdmlIdType‘ objects or ‘NA‘. Publisher identifiers.
experimenter	Experimenter-id-keyed list of ‘experimenterType‘ objects or ‘NA‘.
documentation	Documentation-id-keyed list of ‘documentationType‘ objects or ‘NA‘.
dye	Dye-id-keyed list of ‘dyeType‘ objects or ‘NA‘.
sample	Sample-id-keyed list of ‘sampleType‘ objects or ‘NA‘.
target	Target-id-keyed list of ‘targetType‘ objects or ‘NA‘.
thermalCyclingConditions	Id-keyed list of ‘thermalCyclingConditionsType‘ objects or ‘NA‘.
experiment	Experiment-id-keyed list of ‘experimentType‘ objects or ‘NA‘.

Format

An S7 class.

Value

An S7 object of class ‘rdmlType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘version‘ ‘character(1)‘. RDML schema version represented by this object. Always ‘"1.3"‘ and read-only.

‘dateMade‘ Date-time value or ‘NA‘. Document creation date.

‘dateUpdated‘ Date-time value or ‘NA‘. Last update date.

‘id‘ List of ‘rdmlIdType‘ objects or ‘NA‘. Publisher identifiers.

‘experimenter‘ Experimenter-id-keyed list of ‘experimenterType‘ objects or ‘NA‘.

‘documentation‘ Documentation-id-keyed list of ‘documentationType‘ objects or ‘NA‘.

‘dye‘ Dye-id-keyed list of ‘dyeType‘ objects or ‘NA‘.

‘sample‘ Sample-id-keyed list of ‘sampleType‘ objects or ‘NA‘.

‘target‘ Target-id-keyed list of ‘targetType‘ objects or ‘NA‘.

‘thermalCyclingConditions‘ Id-keyed list of ‘thermalCyclingConditionsType‘ objects or ‘NA‘.

‘experiment‘ Experiment-id-keyed list of ‘experimentType‘ objects or ‘NA‘.

See Also

‘readRDML‘, ‘writeRDML‘, ‘asTable‘, ‘getFData‘, ‘validateRDML‘

`reactPosition` *Return the canonical position of an RDML reaction*

Description

Converts the schema-level reaction identifier to a human-readable plate position using the run PCR format. Conventional alphabetic-row positions always use two-digit columns (for example, ‘A01‘, never ‘A1‘).

Usage

```
reactPosition(react, pcrFormat = NULL)
```

Arguments

`react`	A [reactType] object.
`pcrFormat`	A [pcrFormatType] object, or a missing value when the reaction identifier already contains the position.

Value

A character scalar containing the canonical reaction position.

`reactType` *reactType S7 class*

Description

One reaction/well within a run. It references a sample and stores target-specific qPCR or digital-PCR results. Inherits from ‘rdmlBaseType‘.

Usage

```
reactType(id = NULL, sample = NA, data = NA, partitions = NA)
```

Arguments

`id`	‘idType‘. Reaction identifier or well position.
`sample`	‘idReferenceType‘. Reference to the sample.
`data`	Target-keyed list of ‘dataType‘ objects or ‘NA‘.
`partitions`	List of ‘partitionsType‘ objects or ‘NA‘.

Format

An S7 class.

Value

An S7 object of class ‘reactType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘id‘ ‘idType‘. Reaction identifier or well position.

‘sample‘ ‘idReferenceType‘. Reference to the sample.

‘data‘ Target-keyed list of ‘dataType‘ objects or ‘NA‘.

‘partitions‘ List of ‘partitionsType‘ objects or ‘NA‘.

See Also

‘runType‘, ‘sampleType‘, ‘dataType‘, ‘partitionsType‘

readRDML *Read qPCR data into an RDML object*

Description

Selects a reader from the runtime registry. Readers may return ‘rdmlType‘ directly or ‘rdmlImportData‘, which is built automatically.

Usage

```
readRDML(
  fileName,
  showProgress = TRUE,
  conditionsSep = NULL,
  cluster = NULL,
  format = "auto",
  loss = c("warn", "error", "allow"),
  ...
)
```

Arguments

fileName	Input file.
showProgress	Show importer progress.
conditionsSep	Optional Roche condition separator retained for compatibility.
cluster	Reserved for compatibility.
format	Registered name/alias/extension or ‘"auto"‘.
loss	Loss policy: ‘"warn"‘, ‘"error"‘, or ‘"allow"‘.
...	Format-specific arguments. RDES supports ‘companionFile‘, ‘expId‘, ‘runId‘, and ‘strict‘.

Value

‘rdmlType‘.

See Also

‘writeRDML‘, ‘listRDMLFormats‘, ‘validateRDML‘

registerRDMLFormat *Register a file-format reader and/or writer*

Description

Register a named file format. Automatic dispatch uses the file extension. If several formats support the same extension and operation, the first registered matching format is used by default. Users can select another registered format explicitly with ‘format = "name"‘.

Usage

```
registerRDMLFormat(
  name,
  extensions = character(),
  reader = NULL,
  writer = NULL
)
```

Arguments

name	Unique format name used by ‘format=‘.
extensions	Character vector of file extensions, with or without a leading dot.
reader	Reader function or ‘NULL‘.
writer	Writer function or ‘NULL‘.

Value

Registered format specification invisibly.

See Also

‘loadRDMLModule‘, ‘listRDMLFormats‘, ‘readRDML‘

Examples

```
rdmlFile <- system.file("extdata", "lc96_bACTXY.rdml", package = "RDML7")
demoFile <- tempfile(fileext = ".rdmldemo")
file.copy(rdmlFile, demoFile)

registerRDMLFormat(
  name = "rdml-demo",
  extensions = "rdmldemo",
  reader = function(fileName, ...) {
    readRDML(fileName, format = "rdml", ...)
  }
)
detectRDMLFormat(demoFile, "read")
demo <- readRDML(demoFile, showProgress = FALSE)
```

```
unregisterRDMLFormat("rdml-demo")
unlink(demoFile)
```

---

runType *runType S7 class*

Description

One qPCR instrument run within an experiment. A run contains instrument/acquisition metadata and reactions. Inherits from ‘rdmlBaseType‘.

Usage

```
runType(
  id = NULL,
  description = NA_character_,
  documentation = NA,
  experimenter = NA,
  instrument = NA_character_,
  dataCollectionSoftware = NA,
  backgroundDeterminationMethod = NA_character_,
  cqDetectionMethod = NA,
  thermalCyclingConditions = NA,
  pcrFormat = NA,
  runDate = NA_character_,
  react = NA
)
```

Arguments

id	‘idType‘. Unique run identifier within the experiment.
description	‘character(1)‘ or ‘NA‘. Run description.
documentation	List of ‘idReferenceType‘ objects or ‘NA‘. Documentation references.
experimenter	List of ‘idReferenceType‘ objects or ‘NA‘. Experimenter references.
instrument	‘character(1)‘ or ‘NA‘. Instrument description.
dataCollectionSoftware	‘dataCollectionSoftwareType‘ or ‘NA‘. Acquisition software.
backgroundDeterminationMethod	‘character(1)‘ or ‘NA‘. Background determination method.
cqDetectionMethod	‘cqDetectionMethodType‘ or ‘NA‘. Cq calculation method.
thermalCyclingConditions	‘idReferenceType‘ or ‘NA‘. Reference to thermal cycling conditions.
pcrFormat	‘pcrFormatType‘ or ‘NA‘. Plate/layout format.
runDate	Date-time value or ‘NA‘. Date/time of the run.
react	Reaction-id-keyed list of ‘reactType‘ objects or ‘NA‘.

Format

An S7 class.

Value

An S7 object of class ‘runType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘id‘ ‘idType‘. Unique run identifier within the experiment.

‘description‘ ‘character(1)‘ or ‘NA‘. Run description.

‘documentation‘ List of ‘idReferenceType‘ objects or ‘NA‘. Documentation references.

‘experimenter‘ List of ‘idReferenceType‘ objects or ‘NA‘. Experimenter references.

‘instrument‘ ‘character(1)‘ or ‘NA‘. Instrument description.

‘dataCollectionSoftware‘ ‘dataCollectionSoftwareType‘ or ‘NA‘. Acquisition software.

‘backgroundDeterminationMethod‘ ‘character(1)‘ or ‘NA‘. Background determination method.

‘cqDetectionMethod‘ ‘cqDetectionMethodType‘ or ‘NA‘. Cq calculation method.

‘thermalCyclingConditions‘ ‘idReferenceType‘ or ‘NA‘. Reference to thermal cycling conditions.

‘pcrFormat‘ ‘pcrFormatType‘ or ‘NA‘. Plate/layout format.

‘runDate‘ Date-time value or ‘NA‘. Date/time of the run.

‘react‘ Reaction-id-keyed list of ‘reactType‘ objects or ‘NA‘.

See Also

‘experimentType‘, ‘reactType‘, ‘pcrFormatType‘, ‘thermalCyclingConditionsType‘

`sampleTargetType` *sampleTargetType S7 class*

Description

Sample role, optionally restricted to a target. If ‘targetId‘ is ‘NA‘, the role applies to all targets unless a target-specific role overrides it. Inherits from ‘rdmlBaseType‘.

Usage

```
sampleTargetType(targetId = NA, sampleType = sampleTypeType())
```

Arguments

`targetId`	‘idReferenceType‘ or ‘NA‘. Optional target reference; ‘NA‘ means that the sample role is target-independent.
`sampleType`	‘sampleTypeType‘. Role of the sample for that target.

Format

An S7 class.

Value

An S7 object of class ‘sampleTargetType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘targetId‘ ‘idReferenceType‘ or ‘NA‘. Optional target reference; ‘NA‘ means that the sample role is target-independent.

‘sampleType‘ ‘sampleTypeType‘. Role of the sample for that target.

See Also

‘sampleTypeType‘, ‘sampleType‘, ‘targetType‘

---

sampleType *sampleType S7 class*

Description

A qPCR sample/template solution. Dilutions with different concentrations are represented as different samples; technical replicates normally reference the same sample id, while biological replicates normally use different sample ids. Inherits from ‘rdmlBaseType‘.

Usage

```
sampleType(
  id = NULL,
  description = NA_character_,
  documentation = NA,
  xRef = NA,
  annotation = NA,
  type = NA,
  interRunCalibrator = NA,
  quantity = NA,
  calibratorSample = NA,
  cdnaSynthesisMethod = NA,
  templateQuantity = NA
)
```

Arguments

- `id` ‘idType‘. Unique sample identifier.
- `description` ‘character(1)‘ or ‘NA‘. Sample description.
- `documentation` List of ‘idReferenceType‘ objects or ‘NA‘. References to reusable documentation.
- `xRef` List of ‘xRefType‘ objects or ‘NA‘. External references.
- `annotation` List of ‘annotationType‘ objects or ‘NA‘. Sample annotations.
- `type` List of ‘sampleTargetType‘ objects or ‘NA‘. Roles may be target-independent or target-specific.
- `interRunCalibrator` ‘logical(1)‘ or ‘NA‘. Marks an inter-run calibrator.
- `quantity` Target-keyed list of ‘quantityType‘ objects or ‘NA‘. Target-specific quantities.
- `calibratorSample` ‘logical(1)‘ or ‘NA‘. Marks a calibrator sample.
- `cdnaSynthesisMethod` ‘cdnaSynthesisMethodType‘ or ‘NA‘. cDNA synthesis metadata.
- `templateQuantity` ‘templateQuantityType‘ or ‘NA‘. Template concentration/type.

Format

An S7 class.

Value

An S7 object of class ‘sampleType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘id‘ ‘idType‘. Unique sample identifier.

‘description‘ ‘character(1)‘ or ‘NA‘. Sample description.

‘documentation‘ List of ‘idReferenceType‘ objects or ‘NA‘. References to reusable documentation.

‘xRef‘ List of ‘xRefType‘ objects or ‘NA‘. External references.

‘annotation‘ List of ‘annotationType‘ objects or ‘NA‘. Sample annotations.

‘type‘ List of ‘sampleTargetType‘ objects or ‘NA‘. Roles may be target-independent or target-specific.

‘interRunCalibrator‘ ‘logical(1)‘ or ‘NA‘. Marks an inter-run calibrator.

‘quantity‘ Target-keyed list of ‘quantityType‘ objects or ‘NA‘. Target-specific quantities.

‘calibratorSample‘ ‘logical(1)‘ or ‘NA‘. Marks a calibrator sample.

‘cdnaSynthesisMethod‘ ‘cdnaSynthesisMethodType‘ or ‘NA‘. cDNA synthesis metadata.

‘templateQuantity‘ ‘templateQuantityType‘ or ‘NA‘. Template concentration/type.

See Also

‘sampleTargetType‘, ‘quantityType‘, ‘reactType‘, ‘rdmlType‘

---

sampleTypeType *sampleTypeType S7 enumeration*

Description

Role of a sample for a particular target.

Usage

```
sampleTypeType(value = "")
```

Arguments

`value`	‘character(1)‘. Enumeration value; must be one of the allowed values listed below.

Format

An S7 class.

Value

An S7 object of class ‘sampleTypeType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Allowed values

‘unkn‘ unknown sample.

‘ntc‘ no-template control.

‘nac‘ no-amplification control.

‘std‘ standard.

‘ntp‘ no-target-positive control.

‘nrt‘ no-reverse-transcription control.

‘pos‘ positive control.

‘opt‘ optical calibrator.

See Also

‘sampleTargetType‘, ‘sampleType‘

sequencesType *sequencesType S7 class*

Description

Primer, probe, and amplicon sequences associated with a target. Inherits from ‘rdmlBaseType‘.

Usage

```
sequencesType(
  forwardPrimer = NA,
  reversePrimer = NA,
  probe1 = NA,
  probe2 = NA,
  amplicon = NA
)
```

Arguments

forwardPrimer	‘oligoType‘ or ‘NA‘. Forward primer.
reversePrimer	‘oligoType‘ or ‘NA‘. Reverse primer.
probe1	‘oligoType‘ or ‘NA‘. First probe.
probe2	‘oligoType‘ or ‘NA‘. Second probe.
amplicon	‘oligoType‘ or ‘NA‘. Amplicon sequence.

Format

An S7 class.

Value

An S7 object of class ‘sequencesType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘forwardPrimer‘ ‘oligoType‘ or ‘NA‘. Forward primer.

‘reversePrimer‘ ‘oligoType‘ or ‘NA‘. Reverse primer.

‘probe1‘ ‘oligoType‘ or ‘NA‘. First probe.

‘probe2‘ ‘oligoType‘ or ‘NA‘. Second probe.

‘amplicon‘ ‘oligoType‘ or ‘NA‘. Amplicon sequence.

See Also

‘oligoType‘, ‘targetType‘

`setFData` *Add or replace fluorescence data*

Description

Adds wide fluorescence data and its metadata to an [rdmlType]. Missing experiments, runs, reactions, samples, targets, and dyes are created when enough metadata are supplied.

Usage

```
setFData(x, ...)
```

Arguments

`x`	An [rdmlType] object to modify.
`...`	Reserved for method-specific extensions.

Details

New runs receive a conventional 96-well format, or a 384-well format when numeric reaction identifiers exceed 96. Each fluorescence column must have exactly one matching description row; undescribed columns are skipped with a warning and duplicate descriptions are an error.

RDML objects use value semantics: this function does not modify ‘x‘ in place. Assign the returned value.

Value

The updated [rdmlType] object.

Arguments for the ‘rdmlType‘ method

‘fdata‘ A matrix, data frame, or [data.table::data.table]. The first column contains the coordinate (‘cyc‘ for amplification or ‘tmp‘ for melting); every other fluorescence column is matched to a row of ‘description‘ through ‘fdataName‘. For amplification data, an optional column named ‘tmp‘ is stored as per-cycle temperature unless explicitly described as a fluorescence series.

‘description‘ A metadata table with one row per fluorescence column. Required columns are ‘fdataName‘, ‘expId‘, ‘runId‘, ‘reactId‘, and ‘target‘. ‘sample‘ is required when creating a new reaction, and ‘targetDyeId‘ is required when creating a new target. Recognized optional columns include ‘sampleType‘, ‘targetType‘, ‘cq‘, ‘meltTemp‘, and ‘quantity‘.

‘fdataType‘ Character scalar: ‘"adp"‘ for amplification data or ‘"mdp"‘ for melting data. Defaults to ‘"adp"‘.

‘conflict‘ How to handle metadata that disagree with existing RDML objects: ‘"error"‘ (default) stops, ‘"keep"‘ retains existing metadata, and ‘"replace"‘ updates it. Fluorescence curves supplied in ‘fdata‘ are written for their selected ‘fdataType‘.

See Also

‘buildRDMLFromFData()‘, ‘asTable()‘, ‘getFData()‘

Examples

```
rdmlFile <- system.file("extdata", "lc96_bACTXY.rdml", package = "RDML7")
source <- readRDML(rdmlFile, showProgress = FALSE)
metadata <- asTable(source)
metadata <- metadata[seq_len(min(2L, nrow(metadata))), ]
curves <- getFData(
  source,
  request = metadata,
  dpType = "adp"
)

copied <- setFData(
  rdmlType(),
  fdata = curves,
  description = metadata,
  fdataType = "adp"
)
asTable(copied)
```

---

stepType *stepType S7 class*

Description

One numbered step of a thermal-cycling program. Exactly one of the action properties normally describes the operation performed by the step. Inherits from ‘rdmlBaseType‘.

Usage

```
stepType(
  nr = integer(0),
  description = NA_character_,
  temperature = NA,
  gradient = NA,
  loop = NA,
  pause = NA,
  lidOpen = NA
)
```

Arguments

nr	Positive integer. Step number.
description	‘character(1)‘ or ‘NA‘. Step description.
temperature	‘temperatureType‘ or ‘NA‘. Fixed-temperature action.

`gradient`	‘gradientType‘ or ‘NA‘. Gradient action.
`loop`	‘loopType‘ or ‘NA‘. Loop action.
`pause`	‘pauseType‘ or ‘NA‘. Pause action.
`lidOpen`	‘lidOpenType‘ or ‘NA‘. Lid-open action.

Format

An S7 class.

Value

An S7 object of class ‘stepType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘nr‘ Positive integer. Step number.

‘description‘ ‘character(1)‘ or ‘NA‘. Step description.

‘temperature‘ ‘temperatureType‘ or ‘NA‘. Fixed-temperature action.

‘gradient‘ ‘gradientType‘ or ‘NA‘. Gradient action.

‘loop‘ ‘loopType‘ or ‘NA‘. Loop action.

‘pause‘ ‘pauseType‘ or ‘NA‘. Pause action.

‘lidOpen‘ ‘lidOpenType‘ or ‘NA‘. Lid-open action.

See Also

‘thermalCyclingConditionsType‘, ‘temperatureType‘, ‘gradientType‘, ‘loopType‘

---

summary,RDML7::rdmlType-method

Summarize an RDML object

Description

Provides a compact structural summary of an [rdmlType] object.

Usage

```
## S7 method for class <RDML7::rdmlType>
summary(object, ...)
```

Arguments

`object`	An [rdmlType] object.
`...`	Additional arguments. Currently unused.

Details

‘summary()‘ reports the RDML version and dates together with counts of experiments, runs, reactions, data elements, samples, targets, dyes, thermal cycling conditions, amplification curves (‘adp‘), and melting curves (‘mdp‘).

The historical ‘totalMultiTm‘ field is no longer calculated or returned.

Value

An object of class ‘summary.rdmlType‘.

Examples

```
rdmlFile <- system.file("extdata", "lc96_bACTXY.rdml", package = "RDML7")
rdml <- readRDML(rdmlFile, showProgress = FALSE)
summary(rdml)
```

---

targetType *targetType S7 class*

Description

Description of a qPCR target, including dye, efficiency, sequence, and assay metadata. Inherits from ‘rdmlBaseType‘.

Usage

```
targetType(
  id = NULL,
  description = NA_character_,
  documentation = NA,
  xRef = NA,
  type = NA,
  amplificationEfficiencyMethod = NA_character_,
  amplificationEfficiency = NA_real_,
  amplificationEfficiencySE = NA_real_,
  meltingTemperature = NA_real_,
  detectionLimit = NA_real_,
  dyeId = NA,
  sequences = NA,
  commercialAssay = NA
)
```

Arguments

`id`	‘idType‘. Unique target identifier.
`description`	‘character(1)‘ or ‘NA‘. Target description.
`documentation`	List of ‘idReferenceType‘ objects or ‘NA‘. Documentation references.
`xRef`	List of ‘xRefType‘ objects or ‘NA‘. External references.
`type`	‘targetTypeType‘. Reference target or target of interest.
`amplificationEfficiencyMethod`	‘character(1)‘ or ‘NA‘. Method used to determine amplification efficiency.
`amplificationEfficiency`	‘numeric(1)‘ or ‘NA‘. Amplification efficiency.
`amplificationEfficiencySE`	‘numeric(1)‘ or ‘NA‘. Standard error of amplification efficiency.
`meltingTemperature`	‘numeric(1)‘ or ‘NA‘. Expected target melting temperature.
`detectionLimit`	‘numeric(1)‘ or ‘NA‘. Detection limit.
`dyeId`	‘idReferenceType‘. Reference to the detection dye.
`sequences`	‘sequencesType‘ or ‘NA‘. Primers/probes/amplicon.
`commercialAssay`	‘commercialAssayType‘ or ‘NA‘. Commercial assay metadata.

Format

An S7 class.

Value

An S7 object of class ‘targetType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘id‘ ‘idType‘. Unique target identifier.

‘description‘ ‘character(1)‘ or ‘NA‘. Target description.

‘documentation‘ List of ‘idReferenceType‘ objects or ‘NA‘. Documentation references.

‘xRef‘ List of ‘xRefType‘ objects or ‘NA‘. External references.

‘type‘ ‘targetTypeType‘. Reference target or target of interest.

‘amplificationEfficiencyMethod‘ ‘character(1)‘ or ‘NA‘. Method used to determine amplification efficiency.

‘amplificationEfficiency‘ ‘numeric(1)‘ or ‘NA‘. Amplification efficiency.

‘amplificationEfficiencySE‘ ‘numeric(1)‘ or ‘NA‘. Standard error of amplification efficiency.

‘meltingTemperature‘ ‘numeric(1)‘ or ‘NA‘. Expected target melting temperature.

‘detectionLimit‘ ‘numeric(1)‘ or ‘NA‘. Detection limit.

‘dyeId‘ ‘idReferenceType‘. Reference to the detection dye.

‘sequences‘ ‘sequencesType‘ or ‘NA‘. Primers/probes/amplicon.

‘commercialAssay‘ ‘commercialAssayType‘ or ‘NA‘. Commercial assay metadata.

See Also

‘targetTypeType‘, ‘dyeType‘, ‘dataType‘, ‘rdmlType‘

---

targetTypeType *targetTypeType S7 enumeration*

Description

Functional role of a target.

Usage

```
targetTypeType(value = "")
```

Arguments

`value`	‘character(1)‘. Enumeration value; must be one of the allowed values listed below.

Format

An S7 class.

Value

An S7 object of class ‘targetTypeType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Allowed values

‘ref‘ reference target.

‘toi‘ target of interest.

See Also

‘targetType‘

temperatureBaseType *temperatureBaseType S7 class*

Description

Base class for thermal steps with duration, measurement, ramp, and per-cycle changes. Inherits from ‘rdmlBaseType‘.

Usage

```
temperatureBaseType(
  duration = integer(0),
  temperatureChange = NA_real_,
  durationChange = NA_integer_,
  measure = NA,
  ramp = NA_real_
)
```

Arguments

duration	Positive integer. Step duration.
temperatureChange	‘numeric(1)‘ or ‘NA‘. Temperature increment/decrement per cycle.
durationChange	‘integer(1)‘ or ‘NA‘. Duration increment/decrement per cycle.
measure	‘measureType‘ or ‘NA‘. Fluorescence measurement mode.
ramp	‘numeric(1)‘ or ‘NA‘. Ramp rate.

Format

An S7 class.

Value

An S7 object of class ‘temperatureBaseType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘duration‘ Positive integer. Step duration.

‘temperatureChange‘ ‘numeric(1)‘ or ‘NA‘. Temperature increment/decrement per cycle.

‘durationChange‘ ‘integer(1)‘ or ‘NA‘. Duration increment/decrement per cycle.

‘measure‘ ‘measureType‘ or ‘NA‘. Fluorescence measurement mode.

‘ramp‘ ‘numeric(1)‘ or ‘NA‘. Ramp rate.

See Also

‘temperatureType‘, ‘gradientType‘, ‘measureType‘

temperatureType *temperatureType S7 class*

Description

Fixed-temperature thermal-cycling step. Inherits from ‘temperatureBaseType‘.

Usage

```
temperatureType(
  duration = integer(0),
  temperatureChange = NA_real_,
  durationChange = NA_integer_,
  measure = NA,
  ramp = NA_real_,
  temperature = integer(0)
)
```

Arguments

duration	Positive integer. Step duration.
temperatureChange	‘numeric(1)‘ or ‘NA‘. Temperature increment/decrement per cycle.
durationChange	‘integer(1)‘ or ‘NA‘. Duration increment/decrement per cycle.
measure	‘measureType‘ or ‘NA‘. Fluorescence measurement mode.
ramp	‘numeric(1)‘ or ‘NA‘. Ramp rate.
temperature	‘numeric(1)‘. Step temperature.

Format

An S7 class.

Value

An S7 object of class ‘temperatureType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘temperature‘ ‘numeric(1)‘. Step temperature.

See Also

‘temperatureBaseType‘, ‘stepType‘

templateQuantityType *templateQuantityType S7 class*

Description

Total template concentration together with the template nucleotide type. Inherits from ‘rdmlBaseType‘.

Usage

```
templateQuantityType(conc = integer(0), nucleotide = nucleotideType())
```

Arguments

conc	‘numeric(1)‘. Template concentration.
nucleotide	‘nucleotideType‘. Nucleic-acid type.

Format

An S7 class.

Value

An S7 object of class ‘templateQuantityType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘conc‘ ‘numeric(1)‘. Template concentration.

‘nucleotide‘ ‘nucleotideType‘. Nucleic-acid type.

See Also

‘nucleotideType‘, ‘sampleType‘

thermalCyclingConditionsType *thermalCyclingConditionsType S7 class*

Description

A reusable thermal-cycling program referenced by runs or cDNA-synthesis metadata. Inherits from ‘rdmlBaseType‘.

Usage

```
thermalCyclingConditionsType(
  id = NULL,
  description = NA_character_,
  documentation = NA,
  lidTemperature = NA_real_,
  experimenter = NA,
  step = list()
)
```

Arguments

`id`	‘idType‘. Unique program identifier.
`description`	‘character(1)‘ or ‘NA‘. Program description.
`documentation`	List of ‘idReferenceType‘ objects or ‘NA‘. Documentation references.
`lidTemperature`	‘numeric(1)‘ or ‘NA‘. Heated-lid temperature.
`experimenter`	List of ‘idReferenceType‘ objects or ‘NA‘. Experimenter references.
`step`	List of ‘stepType‘ objects. Ordered program steps.

Format

An S7 class.

Value

An S7 object of class ‘thermalCyclingConditionsType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘id‘ ‘idType‘. Unique program identifier.

‘description‘ ‘character(1)‘ or ‘NA‘. Program description.

‘documentation‘ List of ‘idReferenceType‘ objects or ‘NA‘. Documentation references.

‘lidTemperature‘ ‘numeric(1)‘ or ‘NA‘. Heated-lid temperature.

‘experimenter‘ List of ‘idReferenceType‘ objects or ‘NA‘. Experimenter references.

‘step‘ List of ‘stepType‘ objects. Ordered program steps.

See Also

‘stepType‘, ‘runType‘, ‘cdnaSynthesisMethodType‘

unregisterRDMLFormat *Unregister a file format*

Description

Unregister a file format

Usage

```
unregisterRDMLFormat(name, force = FALSE)
```

Arguments

name	Registered format name.
force	Allow removal of a built-in format.

Value

‘TRUE‘ invisibly when removed.

See Also

‘registerRDMLFormat‘, ‘listRDMLFormats‘

validateRDML *Semantically validate an RDML object*

Description

Checks duplicate keys, dangling references, and fluorescence-curve consistency in addition to S7 property validation.

Usage

```
validateRDML(
  x,
  level = c("full", "structure", "references", "data"),
  action = c("return", "warn", "error")
)
```

Arguments

x	‘rdmlType‘.
level	Validation scope: ‘"structure"‘ checks duplicate identifiers, ‘"references"‘ checks links between RDML objects, ‘"data"‘ checks curve coordinates, and ‘"full"‘ performs all checks.
action	‘"return"‘, ‘"warn"‘, or ‘"error"‘.

Value

A ‘data.table‘ of class ‘rdmlValidation‘ describing issues.

See Also

‘isValidRDML‘, ‘rdmlSummary‘

Examples

```
rdmlFile <- system.file("extdata", "lc96_bACTXY.rdml", package = "RDML7")
rdml <- readRDML(rdmlFile, showProgress = FALSE)

issues <- validateRDML(rdml, level = "full", action = "return")
issues
```

---

writeRDML — *Write an RDML object through the file-format registry*

Description

Write an RDML object through the file-format registry

Usage

```
writeRDML(
  x,
  fileName,
  format = "auto",
  overwrite = FALSE,
  loss = c("warn", "error", "allow"),
  ...
)
```

Arguments

`x`	‘rdmlType‘.
`fileName`	Destination path.
`format`	Registered name/alias/extension or ‘"auto"‘.
`overwrite`	Replace an existing destination.
`loss`	Loss policy: ‘"warn"‘, ‘"error"‘, or ‘"allow"‘.
`...`	Format-specific arguments. RDES supports ‘expId‘, ‘runId‘, and ‘rdesType = "auto"‘, ‘"adp"‘, ‘"mdp"‘, or ‘"both"‘.

Value

Writer-specific result, normally an output path invisibly.

See Also

‘readRDML‘, ‘listRDMLFormats‘, ‘newRDMLLossRecord‘

---

xRefType *xRefType S7 class*

Description

External database or ontology cross-reference. Inherits from ‘rdmlBaseType‘.

Usage

```
xRefType(name = NA_character_, id = NA_character_)
```

Arguments

`name`	‘character(1)‘ or ‘NA‘. Name of the external reference system.
`id`	‘character(1)‘ or ‘NA‘. Identifier in that reference system.

Format

An S7 class.

Value

An S7 object of class ‘xRefType‘ representing the documented RDML schema element. Constructor arguments and object properties describe the element’s structure and meaning.

Properties

‘name‘ ‘character(1)‘ or ‘NA‘. Name of the external reference system.

‘id‘ ‘character(1)‘ or ‘NA‘. Identifier in that reference system.

See Also

‘sampleType‘, ‘targetType‘

Index

annotationType, 4
AsDendrogram *(legacy-api)*, 28
asDendrogram, 4
AsTable *(legacy-api)*, 28
asTable, 5
asXML, 7

buildRDMLFromFData, 8
buildRDMLImport, 9

cdnaSynthesisMethodType, 10
commercialAssayType, 11
cqDetectionMethodType, 11

dataCollectionSoftwareType, 12
dataType, 13
detectRDMLFormat, 15
documentationType, 16
dpAmpCurveType, 16
dpMeltingCurveType, 17
dyeChemistryType, 18
dyeType, 19

editRDML, 20
experimenterType, 20
experimentType, 21

GetFData *(legacy-api)*, 28
getFData, 22
gradientType, 24

idReferenceType, 25
idType, 26
isValidRDML, 26

labelFormatType, 27
legacy-api, 28
lidOpenType, 30
listRDMLFormats, 31
loadRDMLModule, 31
loopType, 32

measureType, 33
MergeRDMLs *(legacy-api)*, 28
mergeRDMLs, 33
metadata-schema-types, 35

newRDMLImportData, 35
newRDMLImportSeries, 37
newRDMLLossRecord, 38
nucleotideType, 39

oligoType, 40

partitionDataType, 41
partitionsType, 42
pauseType, 43
pcrFormatType, 44
primingMethodType, 45

quantityType, 46
quantityUnitType, 47

rdml-experimental-types, 48
rdml-sample-target-types, 48
rdml-thermal-types, 49
rdmlEdit *(legacy-api)*, 28
rdmlIdType, 50
rdmlImportData, 51
rdmlImportSeries, 52
rdmlKeyedList, 52
rdmlType, 53
reactPosition, 54
reactType, 55
readRDML, 56
registerRDMLFormat, 57
runType, 58

sampleTargetType, 59
sampleType, 60
sampleTypeType, 62
sequencesType, 63
SetFData *(legacy-api)*, 28

setFData, 64
stepType, 65
summary
(summary,RDML7::rdmlType-method), 66
summary,RDML7::rdmlType-method, 66

targetType, 67
targetTypeType, 69
temperatureBaseType, 70
temperatureType, 71
templateQuantityType, 72
thermalCyclingConditionsType, 72

unregisterRDMLFormat, 74

validateRDML, 74

writeRDML, 75

xRefType, 76