

# Package ‘chihaya’

July 15, 2025

**Title** Save Delayed Operations to a HDF5 File

**Description** Saves the delayed operations of a DelayedArray to a HDF5 file.

This enables efficient recovery of the DelayedArray's contents in other languages and analysis frameworks.

**Version** 1.9.0

**Date** 2024-09-16

**License** GPL-3

**Depends** DelayedArray

**Imports** methods, Matrix, rhdf5, Rcpp, HDF5Array

**Suggests** BiocGenerics, S4Vectors, BiocSingular, ResidualMatrix,  
BiocStyle, testthat, rmarkdown, knitr

**LinkingTo** Rcpp, Rhdf5lib

**VignetteBuilder** knitr

**BugReports** <https://github.com/ArtifactDB/chihaya-R/issues>

**URL** <https://github.com/ArtifactDB/chihaya-R>

**SystemRequirements** C++17, GNU make

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**biocViews** DataImport, DataRepresentation

**git\_url** <https://git.bioconductor.org/packages/chihaya>

**git\_branch** devel

**git\_last\_commit** cf0c4f5

**git\_last\_commit\_date** 2025-04-15

**Repository** Bioconductor 3.22

**Date/Publication** 2025-07-15

**Author** Aaron Lun [cre, aut]

**Maintainer** Aaron Lun <[infinite.monkeys.with.keyboards@gmail.com](mailto:infinite.monkeys.with.keyboards@gmail.com)>

## Contents

|  |           |
|--|-----------|
| allowExternalSeeds . . . . .                                 | 2         |
| chihaya-utils . . . . .                                      | 3         |
| knownOperations . . . . .                                    | 4         |
| loadDelayed . . . . .  | 6         |
| saveDelayed . . . . .  | 6         |
| saveDelayedObject . . . . .                                  | 7         |
| saveDelayedObject,ANY-method . . . . .                       | 8         |
| saveDelayedObject,array-method . . . . .                     | 9         |
| saveDelayedObject,ConstantArraySeed-method . . . . .         | 10        |
| saveDelayedObject,DelayedAbind-method . . . . .              | 11        |
| saveDelayedObject,DelayedAperm-method . . . . .              | 12        |
| saveDelayedObject,DelayedNaryIsoOp-method . . . . .          | 13        |
| saveDelayedObject,DelayedSetDimnames-method . . . . .        | 13        |
| saveDelayedObject,DelayedSubassign-method . . . . .          | 14        |
| saveDelayedObject,DelayedSubset-method . . . . .             | 15        |
| saveDelayedObject,DelayedUnaryIsoOpStack-method . . . . .    | 16        |
| saveDelayedObject,DelayedUnaryIsoOpWithArgs-method . . . . . | 16        |
| saveDelayedObject,HDF5ArraySeed-method . . . . .             | 17        |
| validate . . . . .   | 18        |
| <b>Index</b>   | <b>20</b> |

---

|                    |                                       |
|--------------------|---------------------------------------|
| allowExternalSeeds | <i>Allow saving of external seeds</i> |
|--------------------|---------------------------------------|

---

## Description

Should external array seeds be saved in [saveDelayed](#)? If FALSE, an error is raised upon encountering external references such as [HDF5ArraySeeds](#). This prevents the creation of delayed objects that cannot be used on different filesystems.

## Usage

```
allowExternalSeeds(allow)
```

## Arguments

|       |   |
|-------|---|
| allow | Logical scalar indicating whether to allow downloads of external seeds. |
|-------|---|

## Value

If allow is not supplied, the current value of this flag is returned.

If allow is supplied, it is used to define the value of this flag, and the *previous* value of the flag is returned.

## Author(s)

Aaron Lun

**Examples**

```
allowExternalSeeds()

a <- allowExternalSeeds(FALSE)
allowExternalSeeds()

# Setting it back
allowExternalSeeds(a)
```

chihaya-utils

*Developer utilities for custom extensions***Description**

Convenience utilities for extending the **chihaya** format with “custom” seeds or operations. These should only be used by package developers.

**Usage**

```
.saveList(file, name, x, parent = NULL, vectors.only = FALSE)

.loadList(file, name, parent = NULL, vectors.only = FALSE)

.labelOperationGroup(file, name, op)

.labelArrayGroup(file, name, arr)

.saveDataset(
  file,
  name,
  x,
  parent = NULL,
  scalar = FALSE,
  optimize.type = FALSE,
  h5type = NULL,
  chunks = NULL
)

.pickArrayType(x)
```

**Arguments**

|      |   |
|------|---|
| file | String containing a path to a file.   |
| name | String containing the name of the object inside the file. This should be a full path from the root of the file, unless parent is provided, in which case it may be the name of the child.                   |
| x    | The object to save. <ul style="list-style-type: none"> <li>• For <code>.pickArrayType</code>, this should be an array-like object.</li> <li>• For <code>.saveList</code>, this should be a list.</li> </ul> |

|                            |   |
|----------------------------|---|
|                            | <ul style="list-style-type: none"> <li>For <code>.saveDataset</code>, this should be a integer, logical, character or double vector or array.</li> </ul>  |
| <code>parent</code>        | String containing the name of the parent containing the child name.   |
| <code>vectors.only</code>  | Logical scalar indicating whether elements of <code>x</code> should be saved and loaded as 1-d arrays rather than seeds.  |
| <code>op</code>            | String containing the name of the delayed operation to use to label the group.  |
| <code>arr</code>           | String containing the name of the delayed array to use to label the group.  |
| <code>scalar</code>        | Logical scalar indicating whether length-1 <code>x</code> should be saved to file as a scalar.  |
| <code>optimize.type</code> | Logical scalar indicating whether to optimize the HDF5 storage type for non-scalar, non-string <code>x</code> .   |
| <code>h5type</code>        | String specifying the HDF5 storage type to use for non-scalar, non-string <code>x</code> , see <a href="#">h5const</a> ("H5T") for possible choices. This is ignored if <code>optimize.type=TRUE</code> .   |
| <code>chunks</code>        | Integer vector of length equal to the number of dimensions of non-scalar <code>x</code> , specifying the chunk dimensions to use. If <code>NULL</code> , this is set to the length of <code>x</code> (if <code>x</code> is a vector) or chosen by <b>HDF5Array</b> (if <code>x</code> is an array). |

### Value

`.saveList` and `.saveScalar` will write `x` to file, returning `NULL` invisibly.

`.labelArrayGroup` and `.labelOperationGroup` will apply the label to the specified group, returning `NULL` invisibly.

`.loadList` will return a list containing the contents of `name`. This is guaranteed to contain only vectors (or fail) if `vectors.only=TRUE`.

`.pickArrayType` will return a string containing the **chihaya** type for an array-like `x`.

### Author(s)

Aaron Lun

---

|                 |   |
|-----------------|---|
| knownOperations | <i>Get or set loaders for operations/arrays</i> |
|-----------------|---|

---

### Description

Get or set loading functions for operations or arrays that were saved into the HDF5 file. This enables third-party packages to modify the **chihaya** framework for their own purposes.

### Usage

```
knownOperations(operations)
```

```
knownArrays(arrays)
```

### Arguments

|                         |   |
|-------------------------|---|
| <code>operations</code> | Named list of loading functions for operations. Each function should accept the same arguments as <a href="#">loadDelayed</a> and return a matrix-like object. Names should match the <code>delayed_operation</code> string used to save the operation to file. |
| <code>arrays</code>     | Named list of loading functions for arrays. Each function should accept the same arguments as <a href="#">loadDelayed</a> and return a matrix-like object. Names should match the <code>delayed_array</code> string used to save the array to file.             |

## Details

This function can be used to modify the loading procedure for existing operations/arrays or to add new loaders for new arrays.

Custom arrays should use a "custom " prefix in the name to ensure that they do not clash with future additions to the **chihaya** specification. If an instance of a custom array contains an **r\_package** scalar string dataset inside its HDF5 group, the string is assumed to hold the name of the package that implements its loading handler; if this package is installed, it will be automatically loaded and used by `loadDelayed`.

Custom operations can be added, but they are not currently supported via `validate`, so it is assumed that such operations will be created outside of `saveDelayed`.

## Value

If `operations` is missing, `customLoadOperations` will return a list of the current custom operations that have been registered with **chihaya**. If `operations` is provided, it is used to define the set of custom operations, and the *previous* set of operations is returned. The same approach is used for arrays in `customLoadArrays`.

## Author(s)

Aaron Lun

## Examples

```
library(HDF5Array)
X <- rsparsematrix(100, 20, 0.1)
Y <- DelayedArray(X)
Z <- log2(Y + 1)

temp <- tempfile(fileext=".h5")
saveDelayed(Z, temp)

# Overriding an existing operation:
ops <- knownOperations()
old_unary <- ops[["unary math"]]
ops[["unary math"]] <- function(file, path) {
  cat("WHEE!\n")
  old_unary(file, path)
}
old <- knownOperations(ops)

# Prints our little message:
loadDelayed(temp)

# Setting it back.
knownOperations(old)
```

---

|             |                             |
|-------------|-----------------------------|
| loadDelayed | <i>Load a DelayedMatrix</i> |
|-------------|-----------------------------|

---

**Description**

Load a [DelayedMatrix](#) object from a location within a HDF5 file.

**Usage**

```
loadDelayed(file, path = "delayed")
```

**Arguments**

|      |   |
|------|---|
| file | String containing a path to a HDF5 file.                                  |
| path | String containing a path inside a HDF5 file containing the DelayedMatrix. |

**Value**

A [DelayedMatrix](#) containing the contents at path.

**Author(s)**

Aaron Lun

**See Also**

[knownOperations](#) and [knownArrays](#), to modify the loading procedure.

**Examples**

```
library(HDF5Array)
X <- rsparsematrix(100, 20, 0.1)
Y <- DelayedArray(X)
Z <- log2(Y + 1)

temp <- tempfile(fileext=".h5")
saveDelayed(Z, temp)
loadDelayed(temp)
```

---

|             |                             |
|-------------|-----------------------------|
| saveDelayed | <i>Save a DelayedMatrix</i> |
|-------------|-----------------------------|

---

**Description**

Save a [DelayedMatrix](#) object to a location within a HDF5 file.

**Usage**

```
saveDelayed(x, file, path = "delayed")
```

**Arguments**

|      |   |
|------|---|
| x    | A <a href="#">DelayedArray</a> object.  |
| file | String containing a path to a HDF5 file. This will be created if it does not yet exist.   |
| path | String containing a path inside a HDF5 file. This should not already exist, though any parent groups should already be constructed. |

**Details**

See the various [saveDelayedObject](#) methods for how each suite of delayed operations is handled. Also see <https://artifactorydb.github.io/chihaya/> for more details on the data layout inside the HDF5 file.

**Value**

The contents of x are written to file and a NULL is invisibly returned.

**Author(s)**

Aaron Lun

**Examples**

```
library(HDF5Array)
X <- rsparsematrix(100, 20, 0.1)
Y <- DelayedArray(X)
Z <- log2(Y + 1)

temp <- tempfile(fileext=".h5")
saveDelayed(Z, temp)
rhdf5::h5ls(temp)
```

---

|                   |                              |
|-------------------|------------------------------|
| saveDelayedObject | <i>Save a delayed object</i> |
|-------------------|------------------------------|

---

**Description**

Saves a delayed object recursively.

**Usage**

```
saveDelayedObject(x, file, name)

## S4 method for signature 'DelayedArray'
saveDelayedObject(x, file, name)
```

**Arguments**

|      |   |
|------|---|
| x    | An R object containing a delayed operation or seed class. |
| file | String containing the path to a HDF5 file.                |
| name | String containing the name of the group to save into.     |

**Details**

The `saveDelayedObject` generic is intended for developers to create methods for new operations. End-users should use the `saveDelayed` function instead.

The `DelayedArray` method will simply extract the seed and use it to call `saveDelayedObject` again.

**Value**

A NULL is returned invisibly. A group is created at name inside file and the delayed operation is saved within.

**Author(s)**

Aaron Lun

**Examples**

```
library(HDF5Array)
X <- rsparsematrix(100, 20, 0.1)
Y <- DelayedArray(X)[1:10,1:5]

temp <- tempfile(fileext=".h5")
rhdf5::h5createFile(temp)
saveDelayedObject(Y, temp, "F00")
rhdf5::h5ls(temp)
```

---

saveDelayedObject, ANY-method

*Saving other seed classes*

---

**Description**

Optional methods to save other classes, depending on the availability of the packages in the current R installation.

**Usage**

```
## S4 method for signature 'ANY'
saveDelayedObject(x, file, name)
```

**Arguments**

|      |   |
|------|---|
| x    | An R object of a supported class, see Details.        |
| file | String containing the path to a HDF5 file.            |
| name | String containing the name of the group to save into. |

## Details

The ANY method will dispatch to classes that are implemented in other packages:

- If `x` is a `LowRankMatrixSeed` from the **BiocSingular** package, it is handled as a delayed matrix product.
- If `x` is a `ResidualMatrixSeed` from the **ResidualMatrix** package, it is converted into the corresponding series of delayed operations. However, the top-level group will contain a `"r_type_hint"` dataset to indicate that it was originally a `ResidualMatrix` object. This provides R clients with the opportunity to reload it as a `ResidualMatrix`, which may be more efficient than the naive `DelayedArray` representation.
- Otherwise, if `x` comes from package **Y**, we will try to load **chihaya.Y**. This is assumed to define an appropriate `saveDelayedObject` method for `x`.

## Value

A `NULL`, invisibly. A group is created at name containing the contents of `x`.

## Author(s)

Aaron Lun

## Examples

```
# Saving a matrix product.
library(BiocSingular)
left <- matrix(rnorm(100000), ncol=20)
right <- matrix(rnorm(50000), ncol=20)
thing <- LowRankMatrix(left, right)
temp <- tempfile()
saveDelayed(thing, temp)
rhdf5::h5ls(temp)
loadDelayed(temp)
```

---

saveDelayedObject,array-method

*Saving simple seed classes*

---

## Description

Methods to save simple seed classes - namely, ordinary matrices or sparse **Matrix** objects - into the delayed operation file. See “Dense arrays” and “Sparse matrices” at <https://artifactorydb.github.io/chihaya/> for more details.

## Usage

```
## S4 method for signature 'array'
saveDelayedObject(x, file, name)

## S4 method for signature 'CsparseMatrix'
saveDelayedObject(x, file, name)
```

**Arguments**

|      |   |
|------|---|
| x    | An R object of the indicated class.                   |
| file | String containing the path to a HDF5 file.            |
| name | String containing the name of the group to save into. |

**Details**

For string arrays, missing values are handled by the "missing-value-placeholder" attribute on the data dataset. All NA values in the array are replaced by the placeholder value in the attribute when they are saved inside the HDF5 file. If this attribute is not present, it can be assumed that all strings are non-missing.

**Value**

A NULL, invisibly. A group is created at name containing the contents of x.

**Author(s)**

Aaron Lun

**Examples**

```
# Saving an ordinary matrix.
X <- matrix(rpois(100, 2), 5, 20)
Y <- DelayedArray(X)
temp <- tempfile(fileext=".h5")
saveDelayed(Y, temp)
rhdf5::h5ls(temp)
loadDelayed(temp)

# Saving a sparse matrix.
X <- rsparsematrix(100, 20, 0.1)
Y <- DelayedArray(X)
temp <- tempfile(fileext=".h5")
saveDelayed(Y, temp)
rhdf5::h5ls(temp)
loadDelayed(temp)
```

---

saveDelayedObject, ConstantArraySeed-method  
*Saving a ConstantArraySeed*

---

**Description**

Save a [ConstantArraySeed](https://artifactorydb.github.io/chihaya/) object. See the "Constant array" section at <https://artifactorydb.github.io/chihaya/> for more details.

**Usage**

```
## S4 method for signature 'ConstantArraySeed'
saveDelayedObject(x, file, name)
```

**Arguments**

|      |   |
|------|---|
| x    | A <a href="#">ConstantArraySeed</a> object.           |
| file | String containing the path to a HDF5 file.            |
| name | String containing the name of the group to save into. |

**Value**

A NULL, invisibly. A group is created at name containing the contents of the ConstantArraySeed.

**Author(s)**

Aaron Lun

**Examples**

```
X <- ConstantArray(value=NA_real_, dim=c(11, 25))
temp <- tempfile(fileext=".h5")
saveDelayed(X, temp)
rhdf5::h5ls(temp)
loadDelayed(temp)
```

---

saveDelayedObject,DelayedAbind-method  
*Saving a DelayedAbind*

---

**Description**

Save a [DelayedAbind](#) object. See the “Combining” operation at <https://artifactorydb.github.io/chihaya/> for more details.

**Usage**

```
## S4 method for signature 'DelayedAbind'
saveDelayedObject(x, file, name)
```

**Arguments**

|      |   |
|------|---|
| x    | A <a href="#">DelayedAbind</a> object.                |
| file | String containing the path to a HDF5 file.            |
| name | String containing the name of the group to save into. |

**Value**

A NULL, invisibly. A group is created at name containing the contents of the DelayedAbind.

**Author(s)**

Aaron Lun

**Examples**

```
X <- DelayedArray(matrix(runif(100), ncol=20))
Y <- cbind(X, X)
temp <- tempfile(fileext=".h5")
saveDelayed(Y, temp)
rhdf5::h5ls(temp)
loadDelayed(temp)
```

---

saveDelayedObject,DelayedAperm-method  
*Saving a DelayedAperm*

---

**Description**

Save a [DelayedAperm](#) object. See the “Transposition” operation at <https://artifacddb.github.io/chihaya/> for more details.

**Usage**

```
## S4 method for signature 'DelayedAperm'
saveDelayedObject(x, file, name)
```

**Arguments**

|      |   |
|------|---|
| x    | A <a href="#">DelayedAperm</a> object.                |
| file | String containing the path to a HDF5 file.            |
| name | String containing the name of the group to save into. |

**Value**

A NULL, invisibly. A group is created at name containing the contents of the DelayedAperm.

**Author(s)**

Aaron Lun

**Examples**

```
X <- DelayedArray(matrix(runif(100), ncol=20))
Y <- t(X)
temp <- tempfile(fileext=".h5")
saveDelayed(Y, temp)
rhdf5::h5ls(temp)
loadDelayed(temp)
```

---

saveDelayedObject,DelayedNaryIsoOp-method  
*Saving a DelayedNaryIsoOp*

---

## Description

Save a [DelayedNaryIsoOp](#) object into a HDF5 file. See the “Binary ...” operations at <https://artifacddb.github.io/chihaya/> for more details.

## Usage

```
## S4 method for signature 'DelayedNaryIsoOp'
saveDelayedObject(x, file, name)
```

## Arguments

|      |   |
|------|---|
| x    | A <a href="#">DelayedNaryIsoOp</a> object.            |
| file | String containing the path to a HDF5 file.            |
| name | String containing the name of the group to save into. |

## Value

A NULL, invisibly. A group is created at name containing the contents of the DelayedNaryIsoOp.

## Author(s)

Aaron Lun

## Examples

```
X <- DelayedArray(matrix(runif(100), ncol=5))
Y <- DelayedArray(matrix(runif(100), ncol=5))
Z <- X * Y
temp <- tempfile(fileext=".h5")
saveDelayed(Z, temp)
rhdf5::h5ls(temp)
loadDelayed(temp)
```

---

saveDelayedObject,DelayedSetDimnames-method  
*Saving a DelayedSetDimnames*

---

## Description

Save a [DelayedSetDimnames](#) object. See the “Dimnames assignment” operation at <https://artifacddb.github.io/chihaya/> for more details.

**Usage**

```
## S4 method for signature 'DelayedSetDimnames'
saveDelayedObject(x, file, name)
```

**Arguments**

|      |   |
|------|---|
| x    | A <a href="#">DelayedSetDimnames</a> object.          |
| file | String containing the path to a HDF5 file.            |
| name | String containing the name of the group to save into. |

**Value**

A NULL, invisibly. A group is created at name containing the contents of the DelayedSetDimnames.

**Author(s)**

Aaron Lun

**Examples**

```
X <- DelayedArray(matrix(runif(100), ncol=20))
colnames(X) <- LETTERS[1:20]
temp <- tempfile(fileext=".h5")
saveDelayed(X, temp)
rhdf5::h5ls(temp)
loadDelayed(temp)
```

---

saveDelayedObject, DelayedSubassign-method  
*Saving a DelayedSubassign*

---

**Description**

Save a [DelayedSubassign](#) object into a HDF5 file. See the “Subset assignment” operation at <https://artifactorydb.github.io/chihaya/> for more details.

**Usage**

```
## S4 method for signature 'DelayedSubassign'
saveDelayedObject(x, file, name)
```

**Arguments**

|      |   |
|------|---|
| x    | A <a href="#">DelayedSubassign</a> object.            |
| file | String containing the path to a HDF5 file.            |
| name | String containing the name of the group to save into. |

**Value**

A NULL, invisibly. A group is created at name containing the contents of the DelayedSubassign.

**Author(s)**

Aaron Lun

**Examples**

```
X <- DelayedArray(matrix(runif(100), ncol=20))
X[1:2,3:5] <- matrix(-runif(6), ncol=3)
temp <- tempfile(fileext=".h5")
saveDelayed(X, temp)
rhdf5::h5ls(temp)
loadDelayed(temp)
```

---

saveDelayedObject,DelayedSubset-method

*Saving a DelayedSubset*


---

**Description**

Save a [DelayedSubset](#) object into a HDF5 file. See the “Subsetting” operation at <https://artifacddb.github.io/chihaya/> for more details.

**Usage**

```
## S4 method for signature 'DelayedSubset'
saveDelayedObject(x, file, name)
```

**Arguments**

|      |   |
|------|---|
| x    | A <a href="#">DelayedSubset</a> object.               |
| file | String containing the path to a HDF5 file.            |
| name | String containing the name of the group to save into. |

**Value**

A NULL, invisibly. A group is created at name containing the contents of the DelayedSubset.

**Author(s)**

Aaron Lun

**Examples**

```
X <- DelayedArray(matrix(runif(100), ncol=20))
Y <- X[1:2,3:5]
temp <- tempfile(fileext=".h5")
saveDelayed(Y, temp)
rhdf5::h5ls(temp)
loadDelayed(temp)
```

---

saveDelayedObject,DelayedUnaryIsoOpStack-method  
*Saving a DelayedUnaryIsoOpStack*

---

## Description

Save a [DelayedUnaryIsoOpStack](#) object into a HDF5 file. See the “Unary ...” operations at <https://artifacddb.github.io/chihaya/> for more details.

## Usage

```
## S4 method for signature 'DelayedUnaryIsoOpStack'
saveDelayedObject(x, file, name)
```

## Arguments

|      |   |
|------|---|
| x    | A <a href="#">DelayedUnaryIsoOpStack</a> object.      |
| file | String containing the path to a HDF5 file.            |
| name | String containing the name of the group to save into. |

## Value

A NULL, invisibly. A group is created at name containing the contents of the DelayedUnaryIsoOpStack.

## Author(s)

Aaron Lun

## Examples

```
X <- DelayedArray(matrix(runif(100), ncol=20))
Y <- log2(X + 10)
temp <- tempfile(fileext=".h5")
saveDelayed(Y, temp)
rhdf5::h5ls(temp)
loadDelayed(temp)
```

---

saveDelayedObject,DelayedUnaryIsoOpWithArgs-method  
*Saving a DelayedUnaryIsoOpWithArgs*

---

## Description

Save a [DelayedUnaryIsoOpWithArgs](#) object into a HDF5 file. See the “Unary ...” operation at <https://artifacddb.github.io/chihaya/> for more details.

**Usage**

```
## S4 method for signature 'DelayedUnaryIsoOpWithArgs'
saveDelayedObject(x, file, name)
```

**Arguments**

|      |   |
|------|---|
| x    | A <a href="#">DelayedUnaryIsoOpWithArgs</a> object.   |
| file | String containing the path to a HDF5 file.            |
| name | String containing the name of the group to save into. |

**Value**

A NULL, invisibly. A group is created at name containing the contents of the DelayedUnaryIsoOpWithArgs.

**Author(s)**

Aaron Lun

**Examples**

```
X <- DelayedArray(matrix(runif(100), ncol=5))
Y <- (1:20 + X) / runif(5)
temp <- tempfile(fileext=".h5")
saveDelayed(Y, temp)
rhdf5::h5ls(temp)
loadDelayed(temp)
```

---

saveDelayedObject,HDF5ArraySeed-method  
*Save HDF5-based seeds*

---

**Description**

Save [HDF5ArraySeed](#) or [H5SparseMatrix](#) objects or their subclasses. See “External HDF5 arrays” at <https://artifactory.github.io/chihaya/> for more details.

**Usage**

```
## S4 method for signature 'HDF5ArraySeed'
saveDelayedObject(x, file, name)

## S4 method for signature 'H5SparseMatrixSeed'
saveDelayedObject(x, file, name)
```

**Arguments**

|      |   |
|------|---|
| x    | A <a href="#">HDF5ArraySeed</a> or <a href="#">H5SparseMatrix</a> object or subclass thereof. |
| file | String containing the path to a HDF5 file.  |
| name | String containing the name of the group to save into.   |

**Value**

A NULL, invisibly. A group is created at name containing the contents of the HDF5-based seed.

**Author(s)**

Aaron Lun

**Examples**

```
library(HDF5Array)
X <- writeHDF5Array(matrix(runif(100), ncol=20))
Y <- X + 1
temp <- tempfile(fileext=".h5")
saveDelayed(Y, temp)
rhdf5::h5ls(temp)
loadDelayed(temp)
```

---

validate

*Validate an artifact*

---

**Description**

Validate the delayed objects inside a HDF5 file. This is automatically run at the end of every `saveDelayed` call to check the integrity of the saved files. See <https://artifactdb.github.io/chihaya/> for more details.

**Usage**

```
validate(path, name)
```

**Arguments**

|      |   |
|------|---|
| path | String containing the path to the HDF5 file.                      |
| name | String containing the name of the delayed object inside the file. |

**Value**

NULL if there are no problems, otherwise an error is raised.

**Author(s)**

Aaron Lun

**See Also**

See <https://artifactdb.github.io/chihaya/> for the specification.

**Examples**

```
X <- DelayedArray(matrix(runif(100), ncol=20))
Y <- X[1:2,3:5]
temp <- tempfile(fileext=".h5")
saveDelayed(Y, temp)
validate(temp, "delayed")
```

# Index

`.labelArrayGroup (chihaya-utils)`, [3](#)  
`.labelOperationGroup (chihaya-utils)`, [3](#)  
`.loadList (chihaya-utils)`, [3](#)  
`.pickArrayType (chihaya-utils)`, [3](#)  
`.saveDataset (chihaya-utils)`, [3](#)  
`.saveList (chihaya-utils)`, [3](#)  
  
`allowExternalSeeds`, [2](#)  
  
`chihaya-utils`, [3](#)  
`ConstantArraySeed`, [10](#), [11](#)  
  
`DelayedAbind`, [11](#)  
`DelayedAperm`, [12](#)  
`DelayedArray`, [7](#), [8](#)  
`DelayedMatrix`, [6](#)  
`DelayedNaryIsoOp`, [13](#)  
`DelayedSetDimnames`, [13](#), [14](#)  
`DelayedSubassign`, [14](#)  
`DelayedSubset`, [15](#)  
`DelayedUnaryIsoOpStack`, [16](#)  
`DelayedUnaryIsoOpWithArgs`, [16](#), [17](#)  
  
`h5const`, [4](#)  
`H5SparseMatrix`, [17](#)  
`HDF5ArraySeed`, [2](#), [17](#)  
  
`knownArrays`, [6](#)  
`knownArrays (knownOperations)`, [4](#)  
`knownOperations`, [4](#), [6](#)  
  
`loadDelayed`, [4](#), [5](#), [6](#)  
  
`saveDelayed`, [2](#), [5](#), [6](#), [8](#), [18](#)  
`saveDelayedObject`, [7](#), [7](#), [8](#)  
`saveDelayedObject, ANY-method`, [8](#)  
`saveDelayedObject, array-method`, [9](#)  
`saveDelayedObject, ConstantArraySeed-method`,  
[10](#)  
`saveDelayedObject, CsparseMatrix-method`  
`(saveDelayedObject, array-method)`,  
[9](#)  
`saveDelayedObject, DelayedAbind-method`,  
[11](#)  
`saveDelayedObject, DelayedAperm-method`,  
[12](#)  
`saveDelayedObject, DelayedArray-method`  
`(saveDelayedObject)`, [7](#)  
`saveDelayedObject, DelayedNaryIsoOp-method`,  
[13](#)  
`saveDelayedObject, DelayedSetDimnames-method`,  
[13](#)  
`saveDelayedObject, DelayedSubassign-method`,  
[14](#)  
`saveDelayedObject, DelayedSubset-method`,  
[15](#)  
`saveDelayedObject, DelayedUnaryIsoOpStack-method`,  
[16](#)  
`saveDelayedObject, DelayedUnaryIsoOpWithArgs-method`,  
[16](#)  
`saveDelayedObject, H5SparseMatrixSeed-method`  
`(saveDelayedObject, HDF5ArraySeed-method)`,  
[17](#)  
`saveDelayedObject, HDF5ArraySeed-method`,  
[17](#)  
  
`validate`, [5](#), [18](#)