

# Package ‘RNifti’

October 2, 2016

**Version** 0.2.2

**Date** 2016-09-29

**Title** Fast R and C++ Access to NIfTI Images

**Imports** Rcpp (>= 0.11.0)

**Suggests** testthat (>= 0.11.0)

**Enhances** oro.nifti, tractor.base

**LinkingTo** Rcpp

**Description** Provides very fast access to images stored in the NIfTI-1 file format <<http://nifti.nimh.nih.gov/nifti-1>>, with seamless synchronisation between compiled C and interpreted R code. Not to be confused with 'RNiftyReg', which provides tools for image registration.

**License** GPL-2

**URL** <https://github.com/jonclayden/RNifti>

**BugReports** <https://github.com/jonclayden/RNifti/issues>

**Encoding** UTF-8

**RoxygenNote** 5.0.1

**NeedsCompilation** yes

**Author** Jon Clayden [cre, aut],  
Bob Cox [aut],  
Mark Jenkinson [aut],  
Rick Reynolds [ctb],  
Kate Fissell [ctb],  
Jean-loup Gailly [cph],  
Mark Adler [cph]

**Maintainer** Jon Clayden <[code@clayden.org](mailto:code@clayden.org)>

**Repository** CRAN

**Date/Publication** 2016-10-02 15:49:31

**R topics documented:**

dim.internalImage . . . . .	2
dumpNifti . . . . .	3
ndim . . . . .	4
pixdim . . . . .	4
readNifti . . . . .	6
retrieveNifti . . . . .	7
updateNifti . . . . .	8
voxelToWorld . . . . .	8
writeNifti . . . . .	9
xform . . . . .	10

<b>Index</b>	<b>12</b>
--------------	-----------

---

dim.internalImage	<i>Internal images</i>
-------------------	------------------------

---

**Description**

An internal image is a simple R object with a few attributes including a pointer to an internal C structure, which contains the full image data. They are used in the package for efficiency, but can be converted to a normal R array using the `as.array` method. Attributes of these objects should not be changed.

**Usage**

```
## S3 method for class 'internalImage'
dim(x)

## S3 replacement method for class 'internalImage'
dim(x) <- value

## S3 method for class 'internalImage'
as.array(x, ...)
```

**Arguments**

<code>x</code>	An "internalImage" object.
<code>value</code>	Not used. Changing the dimensions of an internal image is invalid, and will produce an error.
<code>...</code>	Additional parameters to methods. Currently unused.

**Author(s)**

Jon Clayden <code@clayden.org>

---

`dumpNifti`*Dump the contents of an internal NIFTI-1 object*

---

### Description

This function extracts the contents of an internal NIFTI-1 object into an R list. No processing is done to the elements.

### Usage

```
dumpNifti(image)

## S3 method for class 'niftiHeader'
print(x, ...)
```

### Arguments

<code>image</code>	An image, in any acceptable form (see <a href="#">retrieveNifti</a> ).
<code>x</code>	A "niftiHeader" object.
<code>...</code>	Ignored.

### Value

For `dumpNifti`, a list of class "niftiHeader", with named components corresponding to the elements in a raw NIFTI-1 file.

### Author(s)

Jon Clayden <[code@clayden.org](mailto:code@clayden.org)>

### References

The NIFTI-1 standard (<http://nifti.nimh.nih.gov/nifti-1>).

### Examples

```
dumpNifti(system.file("extdata", "example.nii.gz", package="RNifti"))

# Default header for a standard R array
dumpNifti(array(0L, dim=c(10,10)))
```

---

ndim	<i>Number of dimensions</i>
------	-----------------------------

---

**Description**

This function is shorthand for `length(dim(object))`.

**Usage**

```
ndim(object)
```

**Arguments**

object            An R object.

**Value**

The dimensionality of the object. Objects without a `dim` attribute will produce zero.

**Author(s)**

Jon Clayden <code@clayden.org>

**Examples**

```
ndim(array(0L, dim=c(10,10)))
```

---

pixdim	<i>Pixel dimensions and units</i>
--------	-----------------------------------

---

**Description**

By default, these generic functions return or replace the "pixdim" and "pixunits" attributes of their arguments. These represent the physical step size between pixel or voxel centre points, and the spatial and temporal units that they are given in. The former defaults to 1 in each dimension, if there is no attribute.

**Usage**

```
pixdim(object)

## Default S3 method:
pixdim(object)

pixdim(object) <- value

## Default S3 replacement method:
pixdim(object) <- value

pixunits(object)

## Default S3 method:
pixunits(object)

pixunits(object) <- value

## Default S3 replacement method:
pixunits(object) <- value
```

**Arguments**

<code>object</code>	An R object, generally an image.
<code>value</code>	Numeric vector of pixel dimensions along each axis, or character vector of abbreviated units. For dimensions, a scalar value will be recycled if necessary.

**Value**

`pixdim` returns a numeric vector of pixel dimensions. `pixunits` returns a character vector of length up to two, giving the spatial and temporal unit names.

**Author(s)**

Jon Clayden <code@clayden.org>

**Examples**

```
im <- readNifti(system.file("extdata", "example.nii.gz", package="RNifti"))
pixdim(im)
pixunits(im)
```

---

readNifti	<i>Read a NIFTI-1 format file</i>
-----------	-----------------------------------

---

**Description**

This function reads one or more NIFTI-1 files into R, using the standard NIFTI-1 C library.

**Usage**

```
readNifti(file, internal = FALSE)
```

**Arguments**

file	A character vector of file names.
internal	Logical value. If FALSE (the default), an array of class "niftiImage", containing the image pixel or voxel values, will be returned. If TRUE, the return value will be an object of class "internalImage", which contains only minimal meta-data about the image. Either way, the return value has an attribute which points to a C data structure containing the full image.

**Value**

An array or internal image, with class "niftiImage", and possibly also "internalImage".

**Note**

If the `internal` argument is FALSE (the default), the data type of the image pointer will be set to match one of R's native numeric data types, i.e., 32-bit signed integer or 64-bit double-precision floating-point. In these circumstances the data type reported by the `dumpNifti` function will therefore not, in general, match the storage type used in the file. See also the `datatype` argument to `writeNifti`.

**Author(s)**

Jon Clayden <code@clayden.org>

**References**

The NIFTI-1 standard (<http://nifti.nimh.nih.gov/nifti-1>).

**See Also**

[writeNifti](#)

**Examples**

```
path <- system.file("extdata", "example.nii.gz", package="RNifti")
readNifti(path)
readNifti(path, internal=TRUE)
```

---

retrieveNifti	<i>Obtain an internal NIFTI representation of an object</i>
---------------	---

---

**Description**

This function converts filenames, arrays and other image classes into an object of class "internalImage".

**Usage**

```
retrieveNifti(object)
```

**Arguments**

object            Any suitable object (see Details).

**Details**

If the object has an internal NIFTI pointer, that will be retrieved directly. Otherwise, if it is a string, it will be taken to be a filename. If it looks like a "nifti" object (from package `oro.nifti`), or an "MriImage" object (from package `tractor.base`), a conversion will be attempted. A list will be assumed to be of the form produced by `dumpNifti`. Finally, a numeric array or matrix will be converted using default image parameters.

**Value**

An internal image.

**Author(s)**

Jon Clayden <code@clayden.org>

**See Also**

[readNifti](#), [updateNifti](#)

---

 updateNifti

*Update an internal NIfTI-1 object using a template*


---

### Description

This function adds or updates the internal NIfTI-1 object for an array, using metadata from the template. The dimensions and, if available, pixel dimensions, from the `image` will replace those from the template.

### Usage

```
updateNifti(image, template = NULL)
```

### Arguments

<code>image</code>	A numeric array.
<code>template</code>	An image, in any acceptable form (see <a href="#">retrieveNifti</a> ), or a named list of NIfTI-1 properties like that produced by <a href="#">dumpNifti</a> . The default of <code>NULL</code> will have no effect.

### Value

A copy of the original `image`, with its internal image attribute set or updated appropriately.

### Author(s)

Jon Clayden <[code@clayden.org](mailto:code@clayden.org)>

---

 voxelToWorld

*Transform points between voxel and “world” coordinates*


---

### Description

These functions are used to transform points from dimensionless pixel or voxel coordinates to “real-world” coordinates, typically in millimetres, and back. Actual pixel units can be obtained using the [pixunits](#) function.

### Usage

```
voxelToWorld(points, image, simple = FALSE, ...)
```

```
worldToVoxel(points, image, simple = FALSE, ...)
```



**Arguments**

points	A vector giving the coordinates of a point, or a matrix with one point per row.
image	The image in whose space the points are given.
simple	A logical value: if TRUE then the transformation is performed simply by rescaling the points according to the voxel dimensions recorded in the image. Otherwise the full xform matrix is used.
...	Additional arguments to <a href="#">xform</a> .

**Value**

A vector or matrix of transformed points.

**Note**

Voxel coordinates are assumed by these functions to use R's indexing convention, beginning from 1.

**Author(s)**

Jon Clayden <[code@clayden.org](mailto:code@clayden.org)>

**See Also**

[xform](#), [pixdim](#), [pixunits](#)

**Examples**

```
im <- readNifti(system.file("extdata", "example.nii.gz", package="RNifti"))  
  
# Find the origin  
worldToVoxel(c(0,0,0), im)
```

---

writeNifti

*Write a NIFTI-1 format file*

---

**Description**

This function writes an image to NIFTI-1 format, using the standard NIFTI-1 C library.

**Usage**

```
writeNifti(image, file, template = NULL, datatype = "auto")
```

**Arguments**

image	An image, in any acceptable form (see <a href="#">retrieveNifti</a> ).
file	A character string containing a file name.
template	An optional template object to derive NIFTI-1 properties from. Passed to <a href="#">updateNifti</a> if image is an array.
datatype	The NIFTI datatype to use when writing the data out. The default, "auto" uses the R type or, for internal images, the original datatype. Other possibilities are "float", "int16", etc., which may be preferred to reduce file size. However, no checks are done to ensure that the coercion maintains precision.

**Author(s)**

Jon Clayden <[code@clayden.org](mailto:code@clayden.org)>

**References**

The NIFTI-1 standard (<http://nifti.nimh.nih.gov/nifti-1>).

**See Also**

[readNifti](#), [updateNifti](#)

**Examples**

```
## Not run: writeNifti(image, "image.nii.gz", datatype="float")
```

---

xform

*Obtain or replace the "xform" transforms for an image*

---

**Description**

These functions convert the "qform" or "sform" information in a NIFTI header to or from a corresponding affine matrix. These two "xform" mechanisms are defined by the NIFTI standard, and may both be in use in a particular image header.

**Usage**

```
xform(image, useQuaternionFirst = TRUE)
```

```
qform(x) <- value
```

```
sform(x) <- value
```

**Arguments**

image, x	An image, in any acceptable form (see <a href="#">retrieveNifti</a> ).
useQuaternionFirst	A single logical value. If TRUE, the “qform” matrix will be used first, if it is defined; otherwise the “sform” matrix will take priority.
value	A new 4x4 qform or sform matrix. If the matrix has a “code” attribute, the appropriate qform or sform code is also set.

**Value**

A affine matrix corresponding to the “qform” or “sform” information in the image header. This is a plain matrix, which does not have the “affine” class or source and target attributes.

**Note**

The qform and sform replacement functions are for advanced users only. Modifying the transforms without knowing what you’re doing is usually unwise, as you can make the image object inconsistent.

**Author(s)**

Jon Clayden <[code@clayden.org](mailto:code@clayden.org)>

**References**

The NIfTI-1 standard (<http://nifti.nimh.nih.gov/nifti-1>) is the definitive reference on “xform” conventions.

**Examples**

```
im <- readNifti(system.file("extdata", "example.nii.gz", package="RNifti"))
xform(im)

# Remove the qform information
qform(im) <- structure(diag(4), code=0L)

# The same as above, since the sform is unmodified
xform(im)
```

# Index

as.array.internalImage  
    (dim.internalImage), 2

dim.internalImage, 2  
dim<-.internalImage  
    (dim.internalImage), 2  
dumpNifti, 3, 6–8

internalImage (dim.internalImage), 2

ndim, 4

pixdim, 4, 9  
pixdim<- (pixdim), 4  
pixunits, 8, 9  
pixunits (pixdim), 4  
pixunits<- (pixdim), 4  
print.niftiHeader (dumpNifti), 3

qform<- (xform), 10

readNifti, 6, 7, 10  
retrieveNifti, 3, 7, 8, 10, 11

sform<- (xform), 10

updateNifti, 7, 8, 10

voxelToWorld, 8

worldToVoxel (voxelToWorld), 8  
writeNifti, 6, 9

xform, 9, 10