# Package 'mzID'

November 12, 2026

**Type** Package

**Title** An mzIdentML parser for R

**Version** 1.50.0

**Description** A parser for mzIdentML files implemented using the XML package. The parser tries to be general and able to handle all types of mzIdentML files with the drawback of having less 'pretty' output than a vendor specific parser. Please contact the maintainer with any problems and supply an mzIdentML file so the problems can be fixed quickly.

**License** GPL (>= 2)

**LazyLoad** yes

**biocViews** ImmunoOncology, DataImport, MassSpectrometry, Proteomics

**VignetteBuilder** knitr

**Depends** methods

**Imports** XML, plyr, parallel, doParallel, foreach, iterators, ProtGenerics

**Suggests** knitr, testthat

**Collate** 'aaa.R' 'generics.R' 'mzIDpsm.R' 'mzIDpeptides.R' 'mzIDparameters.R' 'mzIDevidence.R' 'mzIDdatabase.R' 'mzID.R' 'mzIDCollection.R' 'mzID-constructor.R' 'mzID-package.R'

**RoxygenNote** 5.0.1

**git_url** https://git.bioconductor.org/packages/mzID

**git_branch** RELEASE_3_23

**git_last_commit** 2f84f81

**git_last_commit_date** 2026-04-28

**Repository** Bioconductor 3.23

**Date/Publication** 2026-11-11

**Author** Laurent Gatto [ctb, cre] (ORCID: <https://orcid.org/0000-0002-1520-2268>), Thomas Pedersen [aut] (ORCID: <https://orcid.org/0000-0002-6977-7147>), Vladislav Petyuk [ctb]

**Maintainer** Laurent Gatto <laurent.gatto@uclouvain.be>

# Contents

mzID-package . . . 2
flatten . . . 3
increment . . . 4
keyFor . . . 4
mzID . . . 5
mzID-class . . . 6
mzID-getters . . . 9
mzIDCollection . . . 9
mzIDCollection-class . . . 10
mzIDdatabase . . . 13
mzIDdatabase-class . . . 14
mzIDevidence . . . 15
mzIDevidence-class . . . 16
mzIDparameters . . . 17
mzIDparameters-class . . . 18
mzIDpeptides . . . 19
mzIDpeptides-class . . . 20
mzIDpsm . . . 21
mzIDpsm-class . . . 22
removeDecoy . . . 23

**Index** **25**

---

`mzID-package` *A parser for the mzIdentML file format*

---

## Description

Using the `mzID` function this package is able to parse mzIdentML files into an mzID class instance. Multiple files can be parsed in parallel into an mzIDCollection which is a list-like class that handles multiple mzID objects.

## Details

The key functionalities are described in the following pages:

- `mzID`: Parse mzIdentML files
- `mzID-class`: Class to store the data from an mzIdentML file
- `mzIDCollection-class`: Class to store multiple mzID objects
- `mzID-getters`: Access the data stored in mzID and mzIDCollection objects
- `flatten`: Converts the content of an mzID object to a table with respect to the peptide-spectrum matches

The package is maintained at its GitHub repository where feature requests and bugs can be directed. Questions and bugs can furthermore be posted at the Bioconductor support site

### Author(s)

Thomas Lin Pedersen with contributions from Laurent Gatto

---

`flatten` *Flatten an mzID related class into a table*

---

### Description

This function flattens the content of the object into a table by merging the content intelligently (it knows the links between the different objects).

### Usage

```
flatten(object, safeNames = TRUE)

## S4 method for signature 'mzIDpsm'
flatten(object, safeNames = TRUE)

## S4 method for signature 'mzIDpeptides'
flatten(object, safeNames = TRUE)

## S4 method for signature 'mzID'
flatten(object, safeNames = TRUE)

## S4 method for signature 'mzIDCollection'
flatten(object, safeNames = TRUE)
```

### Arguments

| | |
|---|---|
| `object` | The object to be flattened |
| `safeNames` | Logical. Should column names be lowered to ensure compitability between different versions of the mzIdentML schema. Defaults to TRUE |

### Value

A `data.frame` with the flattened result or a list of data.frames

### Methods (by class)

- `mzIDpsm`: Merge id and scans according to the mapping
- `mzIDpeptides`: Merge peptides with their modifications
- `mzID`: Flatten an mzID object with respect to psm'
- `mzIDCollection`: Flatten all mzID object in the collection into a list of data frames.

### See Also

`mzID-class` `mzIDCollection-class` `mzIDpsm` `mzIDpeptides`

## Examples

```
exampleFile <- system.file('extdata', '55merge_tandem.mzid', package = 'mzID')
mzResults <- mzID(exampleFile)
head(flatten(mzResults))
```

---

`increment` *Tools to handle generation of lookup names for the dictionary*

---

## Description

increment looks for a counter in the data environment of mzIDCollection objects and increments it by one if it exists. Otherwise it initialises the counter to 1. It returns the value of the counter.

## Usage

```
increment(object)

## S4 method for signature 'mzIDCollection'
increment(object)
```

## Arguments

| | |
|---|---|
| `object` | An mzIDCollection object |

## Value

An integer given the current count

## Methods (by class)

- `mzIDCollection`: Increment the counter in mzIDcollection objects

---

`keyFor` *Tools to handle generation of lookup names for the dictionary*

---

## Description

keyFor returns the internal hash for a given name.

## Usage

```
keyFor(object, name)

## S4 method for signature 'mzIDCollection,character'
keyFor(object, name)
```

### Arguments

| | |
|---|---|
| object | An mzIDCollection object |

### Value

The hash under wich the object with the given name is stored

### Methods (by class)

- object = mzIDCollection,name = character: Get the key for sample in an mzIDCollection object

---

## mzID *Parse an mzIdentML file*

---

### Description

This function takes a single mzIdentML file and parses it into an mzID object.

### Usage

```
mzID(file, verbose = TRUE)
```

### Arguments

| | |
|---|---|
| file | A character string giving the location of the mzIdentML file to be parsed |
| verbose | Logical Should information be printed to the console? Default is TRUE |

### Details

The mzID function uses the XML package to read the content of an mzIdentML file and store it in an mzID object. Unlike how mzR handles mzML files, mzID parses everything in one chunk. Memory can thus be a problem for very big datasets, but as mzIdentML files are not indexed, it is ineficient to access the data dynamically.

If multiple filenames are passed to the function they will be processed in parallel using foreach and doParallel. The number of workers spawned is either the maximal number of available cores or the number of files to parse, whichever is smallest. The return value will in these cases be an mzIDCollection object. If some of the files cannot be parsed they will not be contained in the returned object and a warning will be issued. No errors will be thrown.

### Value

An mzID object

### See Also

mzID-class mzIDCollection-class

## Examples

```
# Parsing of the example files provided by HUPO:
exampleFiles <- list.files(system.file('extdata', package = 'mzID'),
                           pattern = '*.mzid', full.names = TRUE)
mzID(exampleFiles[1])

mzID(exampleFiles[2])

mzID(exampleFiles[3])

mzID(exampleFiles[4])

mzID(exampleFiles[5])

mzID(exampleFiles[6])

mzID(exampleFiles[7])

mzID(exampleFiles[8])

mzID(exampleFiles[9])

# Parsing into an mzIDCollection
collection <- mzID(exampleFiles[1:3])
names(collection)
```

---

# mzID-class *A class to contain data from mzIdentML-files*

---

## Description

This class stores all parsed information from mzIdentML files

## Usage

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

## S4 method for signature 'mzID'
length(x)

## S4 method for signature 'mzID'
removeDecoy(object)

## S4 method for signature 'mzID'
database(object, safeNames = TRUE)
```

```
## S4 method for signature 'mzID'
evidence(object, safeNames = TRUE)

## S4 method for signature 'mzID'
parameters(object)

## S4 method for signature 'mzID'
software(object)

## S4 method for signature 'mzID'
files(object)

## S4 method for signature 'mzID'
peptides(object, safeNames = TRUE)

## S4 method for signature 'mzID'
modifications(object)

## S4 method for signature 'mzID'
id(object, safeNames = TRUE)

## S4 method for signature 'mzID'
scans(object, safeNames = TRUE)

## S4 method for signature 'mzID'
idScanMap(object)

## S4 method for signature 'mzID'
c(x, y, ..., recursive = FALSE)
```

## Arguments

| | |
|---|---|
| object | An mzID object |
| x | An mzID object |
| safeNames | Should column names be lowercased to ensure compatibility between v1.0 and v1.1 files? |
| y | An mzID or mzIDCollection object |
| ... | ignored |
| recursive | ignored |

## Details

The mzID class stores information in a subset of classes, each class having its own slot. While these classes should not need to be accessed directly, descriptions of their content is delegated to each respective class.

### Methods (by generic)

- show: Short summary of object content
- length: Get number of psm' in object
- removeDecoy: Remove decoys from mzID object
- database: Get the database used for searching
- evidence: Get the evidence from the peptide search
- parameters: Get the parameters used for the search
- software: Get the software used to arrive at the results
- files: Get the data files used for the analysis
- peptides: Get the peptides identified.
- modifications: Get the modification on the identified peptides
- id: Get the identification results
- scans: Get the scans matched to peptides
- idScanMap: Get the link between scans and identifications
- c: Combine mzID and mzIDCollection objects

### Slots

parameters An instance of mzIDparameters-class. This object contains all information related to how the analysis was carried out.

psm An instance of mzIDpsm-class. This object contains the meat of the analysis with all scans and their related PSMs recorded.

peptides An instance of mzIDpeptides-class. This object contains a library of all peptides generated from the database along with possible modifications.

evidence An instance of mzIDevidence-class. This object lists all peptides detected in the analysis with reference to the mzIDpeptides instance.

database An instance of mzIDdatabase-class. This object contains information on the proteins in the database. As the full database is not recorded in mzIdentML files the actual protein sequence is not recorded but there is sufficient information to retrieve it from the database file.

### Objects from the class

Objects can be created using the mzID constructor, which handles parsing of mzIdentML files

### References

http://www.psidev.info/mzidentml

### See Also

mzID

Other mzID.classes: mzIDCollection-class, mzIDdatabase-class, mzIDevidence-class, mzIDparameters-class, mzIDpeptides-class, mzIDpsm-class

## mzID-getters *Getter functions for identification data*

### Description

This set of functions are used to extract data from mzID and mzIDCollection objects.

### Usage

```
evidence(object, safeNames = TRUE)

id(object, safeNames = TRUE)

idScanMap(object)

parameters(object)

software(object)

files(object)
```

### Arguments

| | |
|---|---|
| object | An mzID or mzIDCollection object |
| safeNames | Logical. Should column names be lowered to ensure compitability between different versions of the mzIdentML schema. Defaults to TRUE |

### Value

A data frame or a list of data frames in the case of mzIDCollections

### See Also

mzID-class mzIDCollection-class

## mzIDCollection *Create a new mzIDCollection*

### Description

This function creates a new mzIDCollection object containing the supplied mzID object. As such the result is equivalent to passing a number of mzID objects to `c()`, except that an empty mzIDCollection object is returned if no mzID objects are supplied.

### Usage

```
mzIDCollection(...)
```

### Arguments

| | |
|---|---|
| ... | An arbitrary number of mzID objects |

### Value

An mzIDCollection object

### See Also

mzID-class mzIDCollection-class

---

## mzIDCollection-class *A class to handle a set of mzID objects*

---

### Description

This class is a container for multiple mzID objects. It is constructed such that the bulk data are not copied when passed around. It is the aim that this class have parity with the mzID class in the methods it exposes to the user, such that mzIDCollections can be thought of as vectors in the traditional R sense. Furthermore it accepts standard indexing and concatenation.

### Usage

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

## S4 method for signature 'mzIDCollection'
length(x)

as.list.mzIDCollection(object)

## S4 method for signature 'mzIDCollection'
removeDecoy(object)

## S4 method for signature 'mzIDCollection'
names(x)

## S4 replacement method for signature 'mzIDCollection,character'
names(x) <- value

## S4 method for signature 'mzIDCollection,numeric,missing'
x[[i, j, ...]]
```

```
## S4 method for signature 'mzIDCollection,character,missing'
x[[i, j, ...]]

## S4 method for signature 'mzIDCollection,numeric,missing,missing'
x[i, j, drop]

## S4 method for signature 'mzIDCollection,character,missing,missing'
x[i, j, drop]

## S4 method for signature 'mzIDCollection,logical,missing,missing'
x[i, j, drop]

## S4 method for signature 'mzIDCollection'
c(x, y, ..., recursive = FALSE)

## S4 method for signature 'mzIDCollection'
database(object, safeNames = TRUE)

## S4 method for signature 'mzIDCollection'
evidence(object, safeNames = TRUE)

## S4 method for signature 'mzIDCollection'
parameters(object)

## S4 method for signature 'mzIDCollection'
software(object)

## S4 method for signature 'mzIDCollection'
files(object)

## S4 method for signature 'mzIDCollection'
peptides(object, safeNames = TRUE)

## S4 method for signature 'mzIDCollection'
modifications(object)

## S4 method for signature 'mzIDCollection'
id(object, safeNames = TRUE)

## S4 method for signature 'mzIDCollection'
scans(object, safeNames = TRUE)

## S4 method for signature 'mzIDCollection'
idScanMap(object)
```

## Arguments

| | |
|---|---|
| object | An mzIDCollection object |

| | |
|---|---|
| x | An mzIDCollection object |
| value | A character vector of desired names |
| i | An integer or a string giving the index or the name respectively |
| j | ignored |
| ... | ignored |
| drop | ignored |
| y | An mzID or mzIDCollection object |
| recursive | ignored |
| safeNames | Should column names be lowercased to ensure compatibility between v1.0 and v1.1 files? |

## Details

Objects of this class is usually constructed be passing mulitple files to the mzID constructor, or by combining mulitple mzID objects.

## Methods (by generic)

- show: A short summary of the content of the object
- length: Return the number of mzID object in the collection
- removeDecoy: Removes decoys in all mzID object in collection
- names: Get the names of the mzID object stored in the collection
- names<-: Set the names of the mzID object stored in the collection
- [[: Extract an mzID object by index
- [[: Extract an mzID object by name
- [: Subset collection by index
- [: Subset collection by name
- [: Subset collection by logical value
- c: Combine mzIDCollction and mzID objects
- database: Get the database used for searching
- evidence: Get the evidence from the peptide search
- parameters: Get the parameters used for the search
- software: Get the software used to arrive at the results
- files: Get the data files used for the analysis
- peptides: Get the peptides identified.
- modifications: Get the modification on the identified peptides
- id: Get the identification results
- scans: Get the scans matched to peptides
- idScanMap: Get the link between scans and identifications

## Slots

`data` An environment that holds the individual mzID objects

`.lookup` A matrix with indexing information for retriving the mzID objects in the @data slot.

## See Also

`mzID` `mzIDCollection`

Other mzID.classes: `mzID-class`, `mzIDdatabase-class`, `mzIDevidence-class`, `mzIDparameters-class`, `mzIDpeptides-class`, `mzIDpsm-class`

---

`mzIDdatabase` *A constructor for the mzIDdatabase class*

---

## Description

This function handles parsing of data and construction of an mzIDdatabase object. This function is not intended to be called explicitly but as part of an mzID construction. Thus, the function is not exported.

## Usage

```
mzIDdatabase(doc, ns, addFinalizer = FALSE, path)
```

## Arguments

| | |
|---|---|
| `doc` | an `XMLInternalDocument` created using `xmlInternalTreeParse` |
| `ns` | The appropriate namespace for the doc, as a named character vector with the namespace named x |
| `addFinalizer` | `Logical` Sets whether reference counting should be turned on |
| `path` | If doc is missing the file specified here will be parsed |

## Value

An `mzIDdatabase` object

## See Also

`mzIDdatabase-class`

`mzIDdatabase-class` *A class to store database information from an mzIdentML file*

## Description

This class handles parsing and storage of database information from mzIDentML files, residing at the /MzIdentML/SequenceCollection/DBSequence node.

## Usage

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

## S4 method for signature 'mzIDdatabase'
length(x)

## S4 method for signature 'mzIDdatabase'
database(object, safeNames = TRUE)
```

## Arguments

| | |
|---|---|
| `object` | An mzIDevidence object |
| `x` | An mzIDdatabase object |
| `safeNames` | Should column names be lowercased to ensure compatibility between v1.0 and v1.1 files? |

## Details

The content of the class is stored in a data.frame with columns depending on the content of the mzIdentML file. Required information for files conforming to the mzIdentML standard are: 'accession', 'searchDatabase_ref' and 'id', while additional information can fx be 'length' (number of residues), 'description' (from the fasta file) and 'sequence' (the actual sequence).

## Methods (by generic)

- `show`: Short summary of the content of the object
- `length`: Report the number of proteins in the database
- `database`: Get the database used for searching

## Slots

`database` A data.frame containing references to all the database sequences from the mzIdentML file

### Objects from the class

Objects of mzIDdatabase are not meant to be created explicitly but as part of the mzID-class. Still object can be created with the constructor mzIDdatabase.

### See Also

mzIDdatabase

Other mzID.classes: mzID-class, mzIDCollection-class, mzIDevidence-class, mzIDparameters-class, mzIDpeptides-class, mzIDpsm-class

---

## mzIDevidence *A constructor for the mzIDevidence class*

---

### Description

This function handles parsing of data and construction of an mzIDevidence object. This function is not intended to be called explicitly but as part of an mzID construction. Thus, the function is not exported.

### Usage

```
mzIDevidence(doc, ns, addFinalizer = FALSE, path)
```

### Arguments

| | |
|---|---|
| doc | an XMLInternalDocument created using xmlInternalTreeParse |
| ns | The appropriate namespace for the doc, as a named character vector with the namespace named x |
| addFinalizer | Logical Sets whether reference counting should be turned on |
| path | If doc is missing the file specified here will be parsed |

### Value

An mzIDevidence object

### See Also

mzIDevidence-class

`mzIDevidence-class` *A class to store peptide evidence information from an mzIdentML file*

## Description

This class handles parsing and storage of peptide evidence information from mzIDentML files, residing at the /*/x:SequenceCollection/x:PeptideEvidence node.

## Usage

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

## S4 method for signature 'mzIDevidence'
length(x)

## S4 method for signature 'mzIDevidence'
evidence(object, safeNames = TRUE)
```

## Arguments

| | |
|---|---|
| `object` | An mzIDevidence object |
| `x` | An mzIDevidence object |
| `safeNames` | Should column names be lowercased to ensure compatibility between v1.0 and v1.1 files? |

## Details

The content of the class is stored in a data.frame with columns depending on the content of the mzIdentML file. Columns represent the attribute values of for each PeptideEvidence node. For files conforming to the HUPO standard, dbSequence_ref, id and peptide_ref is required while start, end, pre, post, name, isDecoy, frame and translationTable_ref are optional. Information residing in cvParam and userParam children are not parsed.

## Methods (by generic)

- `show`: Short summary of the content of the object
- `length`: Report number of evidence
- `evidence`: Get the evidence from the peptide search

## Slots

`evidence` A data.frame containing all peptide evidence from the mzIdentML file

### Objects from the class

Objects of mzIDevidence are not meant to be created explicitly but as part of the mzID-class. Still object can be created with the constructor mzIDevidence.

### See Also

mzIDevidence

Other mzID.classes: mzID-class, mzIDCollection-class, mzIDdatabase-class, mzIDparameters-class, mzIDpeptides-class, mzIDpsm-class

---

## mzIDparameters *A constructor for the mzIDparameters class*

---

### Description

This function handles parsing of data and construction of an mzIDparameters object. This function is not intended to be called explicitly but as part of an mzID construction. Thus, the function is not exported. It relies on a number of getter functions to retrive the different information from around the document.

### Usage

```
mzIDparameters(doc, ns, addFinalizer = FALSE, path)
```

### Arguments

| | |
|---|---|
| doc | an XMLInternalDocument created using xmlInternalTreeParse |
| ns | The appropriate namespace for the doc, as a named character vector with the namespace named x |
| addFinalizer | Logical Sets whether reference counting should be turned on |
| path | If doc is missing the file specified here will be parsed |

### Value

An mzIDparameters object

### See Also

mzIDparameters-class

`mzIDparameters-class` *A Class to store analysis information from the mzIdentML file*

## Description

This class tries to collect the multitude of different analysis information required to rerun the analysis. The intended data to be stored are: The software used in the analysis of the data, the location and nature of the rawfile(s), the location and nature of the database file(s), the location of the mzIDentML file itself as well as all the parameters used during the analysis leading to the mzIdentML file. Information regarding how the LC-MS experiment was performed should be collected from the raw data file. As the parameters used in different software solutions can vary greatly, all these parameters are stored in a named list, which can thus be very different from pipeline to pipeline. It is the users responsibility to check conformity between samples.

## Usage

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

## S4 method for signature 'mzIDparameters'
length(x)

## S4 method for signature 'mzIDparameters'
parameters(object)

## S4 method for signature 'mzIDparameters'
software(object)

## S4 method for signature 'mzIDparameters'
files(object)
```

## Arguments

| | |
|---|---|
| `object` | An mzIDparameters object |
| `x` | An mzIDparameters object |

## Methods (by generic)

- `show`: Short summary of the content
- `length`: Get the length of the object
- `parameters`: Get the parameters used for the search
- `software`: Get the software used to arrive at the results
- `files`: Get the data files used for the analysis

## Slots

`software` A data frame with information retaining to the software used for the analysis. At least the name and an id is given, but optionally also version number and URI.

`rawFile` A data frame with information about the raw data file(s) used for the analysis. The data frame will contain at least the location and spectrum ID format.

`databaseFile` A data frame containing at least the location and file format of the database file used in the search.

`idFile` A character string containing the location of the mzIdentML file at the time of parsing.

`parameters` A list containing containing the information stored in the MzIdentML/AnalysisProtocolCollection/SpectrumIde node. SearchType and Threshold are the only required parameters given by the mzIdentML standard.

## Objects from the class

Objects of mzIDparameters are not meant to be created explicitly but as part of the `mzID-class`. Still object can be created with the constructor `mzIDparameters` (not exported).

## See Also

`mzIDparameters`

Other mzID.classes: `mzID-class`, `mzIDCollection-class`, `mzIDdatabase-class`, `mzIDevidence-class`, `mzIDpeptides-class`, `mzIDpsm-class`

---

`mzIDpeptides` *A constructor for the mzIDpeptides class*

---

## Description

This function handles parsing of data and construction of an mzIDpeptides object. This function is not intended to be called explicitly but as part of an mzID construction. Thus, the function is not exported.

## Usage

```
mzIDpeptides(doc, ns, addFinalizer = FALSE, path)
```

## Arguments

| | |
|---|---|
| `doc` | an `XMLInternalDocument` created using `xmlInternalTreeParse` |
| `ns` | The appropriate namespace for the doc, as a named character vector with the namespace named x |
| `addFinalizer` | `Logical` Sets whether reference counting should be turned on |
| `path` | If doc is missing the file specified here will be parsed |

## Value

An mzIDpeptides object

## See Also

mzIDpeptides-class

---

mzIDpeptides-class *A class to store peptide information from an mzIdentML file*

---

## Description

This class handles parsing and storage of peptide information from mzIDentML files, residing at the /x:MzIdentML/x:SequenceCollection/x:Peptide node.

## Usage

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

## S4 method for signature 'mzIDpeptides'
length(x)

## S4 method for signature 'mzIDpeptides'
peptides(object, safeNames = TRUE)

## S4 method for signature 'mzIDpeptides'
modifications(object)
```

## Arguments

| | |
|---|---|
| object | An mzIDpeptides object |
| x | An mzIDpeptides object |
| safeNames | Should column names be lowercased to ensure compatibility between v1.0 and v1.1 files? |

## Details

The information is stored in a dataframe with an id, an optinal name and the amino acid sequence of the peptide. Alongside a list is stored with modification information of each peptide. Each row in the dataframe has a corresponding entry en the list. If no modification of the peptide is present the entry is NULL, if a modification is present the entry is a dataframe, listing the different modifications of the peptide.

### Methods (by generic)

- show: Short summary of the content of the object
- length: Report the number of peptides
- peptides: Get the peptides identified.
- modifications: Get the modification on the identified peptides

### Slots

peptides A data.frame containing all peptides used in the search

modifications A list containing possible modifications of the peptides listed in @peptides

### Objects from the class

Objects of mzIDpeptides are not meant to be created explicitly but as part of the mzID-class. Still object can be created with the constructor mzIDpeptides.

### See Also

mzIDpeptides

Other mzID.classes: mzID-class, mzIDCollection-class, mzIDdatabase-class, mzIDevidence-class, mzIDparameters-class, mzIDpsm-class

---

## mzIDpsm *A constructor for the mzIDpsm class*

---

### Description

This function handles parsing of data and construction of an mzIDpsm object. This function is not intended to be called explicitly but as part of an mzID construction. Thus, the function is not exported.

### Usage

```
mzIDpsm(doc, ns, addFinalizer = FALSE, path)
```

### Arguments

| | |
|---|---|
| doc | an XMLInternalDocument created using xmlInternalTreeParse |
| ns | The appropriate namespace for the doc, as a named character vector with the namespace named x. |
| addFinalizer | Logical Sets whether reference counting should be turned on. |
| path | If doc is missing the file specified here will be parsed. |

### Value

An mzIDpsm object

## See Also

mzIDpsm-class

---

mzIDpsm-class *A class to store psm information from an mzIdentML file*

---

## Description

This class handles parsing and storage of scan info and the related psm's. This information resides in the /*/x:DataCollection/x:AnalysisData/x:SpectrumIdentificationList/x:SpectrumIdentificationResult node.

## Usage

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

## S4 method for signature 'mzIDpsm'
length(x)

## S4 method for signature 'mzIDpsm'
id(object, safeNames = TRUE)

## S4 method for signature 'mzIDpsm'
scans(object, safeNames = TRUE)

## S4 method for signature 'mzIDpsm'
idScanMap(object)
```

## Arguments

| | |
|---|---|
| object | An mzIDpsm object |
| x | An mzIDpsm object |
| safeNames | Should column names be lowercased to ensure compatibility between v1.0 and v1.1 files? |

## Details

The content of the class is stored as two data frames: One containing a row for each scan in the results, and one containing all psm's in the results. Additionally a list containing indexing from scan to psm is stored.

### Methods (by generic)

- `show`: A short summary of the content
- `length`: Get the number of psm'
- `id`: Get the identification results
- `scans`: Get the scans matched to peptides
- `idScanMap`: Get the link between scans and identifications

### Slots

`scans` A data.frame containing all reference to all scans with at least one psm. The columns gives at least an ID, a spectrumID and a reference to the file used.

`id` A data.frame containing all psm's from the analysis. The columns depend on the file but at least id, chargeState, experimentalMassToCharge, passThreshold and rank must exist according to the mzIdentML specifications.

`mapping` A list with an entry for each row in @scans. Each entry contains an integer vector pointing to the related rows in @id.

### Objects from the class

Objects of mzIDpsm are not meant to be created explicitly but as part of the `mzID-class`. Still object can be created with the constructor `mzIDpsm`.

### See Also

`mzIDpsm`

Other mzID.classes: `mzID-class`, `mzIDCollection-class`, `mzIDdatabase-class`, `mzIDevidence-class`, `mzIDparameters-class`, `mzIDpeptides-class`

---

## `removeDecoy` *Remove decoy identification*

---

### Description

This function trims down an mzID or mzIDCollection object by removing all information that is only related to the decoy database search. If some information relates to both the regular and decoy database (e.g. a peptide sequence that can be found in both databases) it is kept.

### Usage

```
removeDecoy(object, ...)
```

### Arguments

| | |
|---|---|
| `object` | An mzID or mzIDCollection to remove decoy information from |
| `...` | Currently ignored |

## Value

An mzID or mzIDCollection object depending on the input

## See Also

mzID-class mzIDCollection-class

# Index


∗ **internal**
    increment, 4
    keyFor, 4
[,mzIDCollection,character,missing,missing-method
        *(mzIDCollection-class)*, 10
[,mzIDCollection,logical,missing,missing-method
        *(mzIDCollection-class)*, 10
[,mzIDCollection,numeric,missing,missing-method
        *(mzIDCollection-class)*, 10
[[,mzIDCollection,character,missing-method
        *(mzIDCollection-class)*, 10
[[,mzIDCollection,numeric,missing-method
        *(mzIDCollection-class)*, 10

as.list.mzIDCollection
        *(mzIDCollection-class)*, 10

c,mzID-method *(mzID-class)*, 6
c,mzIDCollection-method
        *(mzIDCollection-class)*, 10

database,mzID-method *(mzID-class)*, 6
database,mzIDCollection-method
        *(mzIDCollection-class)*, 10
database,mzIDdatabase-method
        *(mzIDdatabase-class)*, 14

evidence *(mzID-getters)*, 9
evidence,mzID-method *(mzID-class)*, 6
evidence,mzIDCollection-method
        *(mzIDCollection-class)*, 10
evidence,mzIDevidence-method
        *(mzIDevidence-class)*, 16

files *(mzID-getters)*, 9
files,mzID-method *(mzID-class)*, 6
files,mzIDCollection-method
        *(mzIDCollection-class)*, 10
files,mzIDparameters-method
        *(mzIDparameters-class)*, 18
flatten, *2*, 3
flatten,mzID-method *(flatten)*, 3
flatten,mzIDCollection-method
        *(flatten)*, 3
flatten,mzIDpeptides-method *(flatten)*, 3
flatten,mzIDpsm-method *(flatten)*, 3

id *(mzID-getters)*, 9
id,mzID-method *(mzID-class)*, 6
id,mzIDCollection-method
        *(mzIDCollection-class)*, 10
id,mzIDpsm-method *(mzIDpsm-class)*, 22
idScanMap *(mzID-getters)*, 9
idScanMap,mzID-method *(mzID-class)*, 6
idScanMap,mzIDCollection-method
        *(mzIDCollection-class)*, 10
idScanMap,mzIDpsm-method
        *(mzIDpsm-class)*, 22
increment, 4
increment,mzIDCollection-method
        *(increment)*, 4

keyFor, 4
keyFor,mzIDCollection,character-method
        *(keyFor)*, 4

length,mzID-method *(mzID-class)*, 6
length,mzIDCollection-method
        *(mzIDCollection-class)*, 10
length,mzIDdatabase-method
        *(mzIDdatabase-class)*, 14
length,mzIDevidence-method
        *(mzIDevidence-class)*, 16
length,mzIDparameters-method
        *(mzIDparameters-class)*, 18
length,mzIDpeptides-method
        *(mzIDpeptides-class)*, 20
length,mzIDpsm-method *(mzIDpsm-class)*,
        22

modifications,mzID-method *(mzID-class)*,
        6

modifications,mzIDCollection-method
(mzIDCollection-class), 10
modifications,mzIDpeptides-method
(mzIDpeptides-class), 20
mzID, 2, 5, 8, *13*
mzID-class, 6
mzID-getters, 9
mzID-package, 2
mzIDCollection, 9, *13*
mzIDCollection-class, 10
mzIDdatabase, 13, *15*
mzIDdatabase-class, 14
mzIDevidence, 15, *17*
mzIDevidence-class, 16
mzIDparameters, 17, *19*
mzIDparameters-class, 18
mzIDpeptides, *3*, 19, *21*
mzIDpeptides-class, 20
mzIDpsm, *3*, 21, *23*
mzIDpsm-class, 22

names,mzIDCollection-method
(mzIDCollection-class), 10
names<-,mzIDCollection,character-method
(mzIDCollection-class), 10

parameters (mzID-getters), 9
parameters,mzID-method (mzID-class), 6
parameters,mzIDCollection-method
(mzIDCollection-class), 10
parameters,mzIDparameters-method
(mzIDparameters-class), 18
peptides,mzID-method (mzID-class), 6
peptides,mzIDCollection-method
(mzIDCollection-class), 10
peptides,mzIDpeptides-method
(mzIDpeptides-class), 20

removeDecoy, 23
removeDecoy,mzID-method (mzID-class), 6
removeDecoy,mzIDCollection-method
(mzIDCollection-class), 10

scans,mzID-method (mzID-class), 6
scans,mzIDCollection-method
(mzIDCollection-class), 10
scans,mzIDpsm-method (mzIDpsm-class), 22
show,mzID-method (mzID-class), 6
show,mzIDCollection-method
(mzIDCollection-class), 10
show,mzIDdatabase-method
(mzIDdatabase-class), 14
show,mzIDevidence-method
(mzIDevidence-class), 16
show,mzIDparameters-method
(mzIDparameters-class), 18
show,mzIDpeptides-method
(mzIDpeptides-class), 20
show,mzIDpsm-method (mzIDpsm-class), 22
software (mzID-getters), 9
software,mzID-method (mzID-class), 6
software,mzIDCollection-method
(mzIDCollection-class), 10
software,mzIDparameters-method
(mzIDparameters-class), 18

xmlInternalTreeParse, *13*, *15*, *17*, *19*, *21*