# Package ‘tm.plugin.alceste’

May 8, 2026

**Type** Package

**Title** Import Texts from Files in the 'Alceste' Format Using the 'tm' Text Mining Framework

**Version** 1.1.2

**Date** 2025-02-27

**Imports** NLP, tm (>= 0.6)

**Suggests** stringi

**Description** Provides a 'tm' Source to create corpora from a corpus prepared in the format used by the 'Alceste' application (i.e. a single text file with inline meta-data). It is able to import both text contents and meta-data (starred) variables.

**License** GPL (>= 2)

**URL** https://github.com/nalimilan/R.TeMiS

**BugReports** https://github.com/nalimilan/R.TeMiS/issues

**NeedsCompilation** no

**Author** Milan Bouchet-Valat [aut, cre]

**Maintainer** Milan Bouchet-Valat <nalimilan@club.fr>

**Repository** CRAN

**Date/Publication** 2025-02-28 09:50:02 UTC

## Contents


tm.plugin.alceste-package . . . . . 2
AlcesteSource . . . . . 3
readAlceste . . . . . 4

**Index** **5**

`tm.plugin.alceste-package` *A plug-in for the tm text mining framework to import corpora from Alceste files*

## Description

This package provides a tm Source to create corpora from files formatted in the format used by the Alceste application.

## Details

Typical usage is to create a corpus from an Alceste file prepared manually (here called `myAlcesteCorpus.txt`). Frequently, it is necessary to specify the encoding of the texts via `link{AlcesteSource}`'s `encoding` argument.

```
# Import corpus
source <- europresseSource("myAlcesteCorpus.txt")
corpus <- Corpus(source)

# See how many articles were imported
corpus

# See the contents of the first article and its meta-data
inspect(corpus[1])
meta(corpus[[1]])
```

See `link{AlcesteSource}` for more details and real examples.

## Author(s)

Milan Bouchet-Valat <nalimilan@club.fr>

## References

https://image-zafar.com/Logicieluk.html

AlcesteSource *Alceste Source*

## Description

Construct a source for an input containing a set of texts saved in the Alceste format in a single text file.

## Usage

```
AlcesteSource(x, encoding = "auto")
```

## Arguments

| | |
|---|---|
| x | Either a character identifying the file or a connection. |
| encoding | A character string: if non-empty declares the encoding used when reading the file, so the character data can be re-encoded. See the 'Encoding' section of the help for `file`. The default, "auto", uses `stri_enc_detect` to try to guess the encoding; this may fail, in which case the native encoding is used. |

## Details

Several texts are saved in a single Alceste-formatted file, separated by lines starting with "***" or digits, followed by starred variables (see links below). These variables are set as document metadata that can be accessed via the `meta` function.

Currently, "theme" lines starting with "-*" are ignored.

## Value

An object of class `AlcesteSource` which extends the class `Source` representing set of articles from Alceste.

## Author(s)

Milan Bouchet-Valat

## See Also

https://image-zafar.com/sites/default/files/telechargements/formatage_alceste.pdf (in French) about the Alceste format

`readAlceste` for the function actually parsing individual articles.

`getSources` to list available sources.

## Examples

```
library(tm)
file <- system.file("texts", "alceste_test.txt",
                    package = "tm.plugin.alceste")
corpus <- Corpus(AlcesteSource(file))

# See the contents of the documents
inspect(corpus)

# See meta-data associated with first article
meta(corpus[[1]])
```

---

`readAlceste` *Read in a text in the Alceste format*

---

## Description

Read in a text in the Alceste format using starred variables.

## Usage

```
readAlceste(elem, language, id)
```

## Arguments

| | |
|---|---|
| `elem` | A `list` with the named element `content` which must hold the document to be read in. |
| `language` | A `character` vector giving the text's language. If set to `NA`, the language will automatically be set to the value reported in the document (which is usually correct). |
| `id` | A `character` vector representing a unique identification string for the returned text document. |

## Value

A `PlainTextDocument` with the contents of the article and the available meta-data set.

## Author(s)

Milan Bouchet-Valat

## See Also

`getReaders` to list available reader functions.

# Index

AlcesteSource, 3

eoi.AlcesteSource *(*AlcesteSource*)*, 3

file, *3*

getElem.AlcesteSource *(*AlcesteSource*)*, 3
getReaders, *4*
getSources, *3*

meta, *3*

readAlceste, *3*, 4

stri_enc_detect, *3*

tm.plugin.alceste
*(*tm.plugin.alceste-package*)*, 2
tm.plugin.alceste-package, 2