

Package ‘angstromATE’

September 24, 2026

Type Package

Title Imports Recipe and Log Files from Angstrom Engineering Thermal Evaporator

Version 0.2.3

Author Thomas Gredig [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0002-5824-7626>>)

Maintainer Thomas Gredig <tgredig@csulb.edu>

Description Imports and parses recipe and log files generated by Angstrom Engineering thermal evaporation systems. These files contain time-series measurements and process metadata, including chamber pressure, source activity, deposition rate, film thickness, tooling parameters, substrate conditions, and related deposition variables. The package provides functions to extract, organize, and summarize process information in a compact format suitable for data analysis, visualization, and quality control workflows.

License GPL (>= 3)

BugReports <https://github.com/thomasgredig/angstromATE/issues>

Encoding UTF-8

Imports stringr, utils, xml2

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

URL <https://github.com/thomasgredig/angstromATE>

NeedsCompilation no

Repository CRAN

Date/Publication 2026-09-24 20:30:02 UTC

Contents

ATE.complete	2
ATE.import	3
ATE.info	3
ATE.readRecipe	4
ATE.sampleFiles	5
ATE.status	6
conv2seconds	7
Index	8

ATE.complete	<i>ATE Deposition Summary</i>
--------------	-------------------------------

Description

Reads an XML Status file from the ATE thermal evaporator and returns the procedural timeline.

Usage

```
ATE.complete(filename, summaryOnly = FALSE)
```

Arguments

filename	XML Status file from Angstrom Thermal Evaporator
summaryOnly	logical, if TRUE, returns summary for one layer

Details

Extracts information about the deposition thickness from the completed status XML file at the end of the deposition.

Value

deposition thickness, rate, ramping times, and actions

Author(s)

Thomas Gredig

Examples

```
fileName <- ATE.sampleFiles('_Complete_')
ATE.complete(fileName, TRUE)
ATE.complete(fileName)
```

`ATE.import`*Imports Angstrom Engineering Thermal Evaporator Log Data*

Description

Imports Angstrom Engineering Thermal Evaporator Log Data

Usage

```
ATE.import(filename)
```

Arguments

filename CSV filename including path for the ATE log file

Value

data frame with around 50 variables and rows that represent time; the variables include Date, Time, SubstrateShutterOpen, ChamberPressure and many other parameters.

Author(s)

Thomas Gredig

Examples

```
fileName = ATE.sampleFiles("csv")[1]
d = ATE.import(fileName)
head(d)
```

`ATE.info`*Basic Information from Thermal Evaporator Deposition*

Description

Imports data from the CSV log file of an Angstrom Engineering Thermal Evaporator. It extracts information during the deposition; i.e. while the shutter is open. It returns a condensed version of the deposition parameters.

Usage

```
ATE.info(filename, verbose = FALSE)
```

Arguments

filename	full path of ATE Log file
verbose	set to TRUE to get additional information

Value

list with information during the deposition that includes the thickness, the deposition time in seconds, the starting date, the substrate heater temperature, the material deposition temperature, tooling factor, base pressure, the pressure at start of the deposition, maximum pressure, and the material name

Author(s)

Thomas Gredig

See Also

[ATE.import()]

Examples

```
fileName = ATE.sampleFiles("csv")[1]
d = ATE.info(fileName,TRUE)
head(d)
```

ATE.readRecipe	<i>Read an ATE .rcp recipe file</i>
----------------	-------------------------------------

Description

Reads an XML-based ATE ‘.rcp’ recipe file and extracts deposition-relevant parameters into a data frame. The returned table contains one row per “Deposit Rate” action in the recipe, preserving recipe order.

Usage

```
ATE.readRecipe(filename)
```

Arguments

filename	Character string. Path to the ‘.rcp’ XML recipe file.
----------	---

Details

For each deposition action, the function reports the source name, the first chamber pressure set point encountered before or within the recipe, the most recent substrate temperature ramp set point before deposition, the target deposition rate, timeout, and target thickness.

XML namespaces are ignored when matching element names.

Source names are resolved by matching each deposition action's 'RecipeSourceID' to the corresponding 'RecipeSource' entry.

The substrate temperature is inferred from the last preceding "Ramp Temperature" action. The function tries the following XML field names, in order, and returns the first numeric value found: 'TemperatureSetpoint', 'TargetTemperature', 'Temperature', and 'Setpoint'.

Value

A 'data.frame' with columns:

source_name Name of the recipe source used for the deposition action.

pre_vacuum_pressure First chamber pressure set point from a "Wait for Chamber Pressure" action.

last_substrate_temperature Most recent substrate temperature ramp target or set point before the deposition action.

deposition_rate Target deposition rate from the "Deposit Rate" action.

deposition_time Timeout value from the "Deposit Rate" action.

thickness Target thickness from the "Deposit Rate" action.

If no deposition actions are found, a one-row data frame is returned with 'NA' values for deposition-specific fields.

Examples

```
## Not run:
recipe <- ATE.readRecipe("my_recipe.rcp")
recipe

## End(Not run)
```

ATE.sampleFiles

Sample ATE file list

Description

Returns a list of sample thermal evaporator log files, mostly for testing.

Usage

```
ATE.sampleFiles(filePattern = "*")
```

Arguments

filePattern pattern to limit the files

Value

list of sample data files with log information

Examples

```
ATE.sampleFiles()  
ATE.sampleFiles('_Status')
```

ATE.status

ATE Recipe Status

Description

Reads an XML Status file from the ATE thermal evaporator and returns the procedural timeline.

Usage

```
ATE.status(filename)
```

Arguments

filename path and filename of XML Status file from Angstrom Thermal Evaporator

Value

data frame with description steps, start and end times

Author(s)

Thomas Gredig

Examples

```
fileName <- ATE.sampleFiles('_Status')  
ATE.status(fileName)
```

conv2seconds	<i>Convert Time String to Numeric</i>
--------------	---------------------------------------

Description

Convert Time String to Numeric

Usage

```
conv2seconds(strTime)
```

Arguments

strTime	a string with time
---------	--------------------

Value

a numeric value in units of seconds

Author(s)

Thomas Gredig

Examples

```
conv2seconds("00:35:40.1816298")  
conv2seconds("00:35:40.1816298") - conv2seconds("00:36:40.1816298")  
conv2seconds("1.19:07:06.5180408")
```

Index

ATE.complete, [2](#)
ATE.import, [3](#)
ATE.info, [3](#)
ATE.readRecipe, [4](#)
ATE.sampleFiles, [5](#)
ATE.status, [6](#)

conv2seconds, [7](#)