

---

# **Micress API Documentation**

***Release 1.0.0***

**R. Altenfeld**

January 19, 2017

---

## micress package

---

### Submodules

#### micress.MICFieldID module

Additional test case and MICRESS specific field IDs

#### micress.MICPropertyID module

Additional test case and MICRESS specific property IDs

#### micress.Property module

**class** micress.Property.**Property** (*value, propID, valueType, time, units, objectID=0*)

Bases: mupif.Property.Property

Extension of the Mupif's property class

**getUnits** ()

The unit string representation is determined by the property ID.

**Returns** unit's representation

**Return type** string

#### micress.micress module

**class** micress.micress.**micress** (*file='', workdir=''*)

Bases: mupif.Application.Application

MICRESS application interface class This class implements the methods of the Mupif Application class for MICRESS.

**getApplicationSignature** ()

Get the interface and external application signature.

**Returns** returns application signature

**Return type** string

**getField** (*fieldID, time*)

This method returns a Mupif field of the MICRESS results according to the given 'fieldID' and 'time'.

**Note** This implementation of the interface supports only MICRESS results written in the VTK format.

**Parameters**

- **fieldID** (*Mupif.FieldID*) – ID of requested field, e.g. FID\_Phase
- **time** (*double*) – simulation time

**Returns** result field in Mupif field format

**Return type** Mupif.Field

**getProperty** (*propID, time, objectID=0*)

This method provides the property value for the given property ID and time. It restores these values from interface's internal data structures. The parsing of the external result files is done in the private parse methods, e.g. `__parseLogfile`. This allows to grab several results with one file parsing step, only.

**Parameters**

- **propID** (*PropertyID*) – property ID, e.g. PID\_Concentration
- **time** (*double*) – simulation time

**Returns** Property object with value set

**Return type** *Property*

**setProperty** (*property, objectID=0*)

This method stores the given property value to the interface's internal data attribute.

**Note** All following actions, like set/get/solveStep, are relative to the property PID\_RVELocation. This has to be set first to act on the appropriate MICRESS simulation results.

**Parameters** **property** (*Mupif.Property*) – the property to set

**solveStep** (*tstep, stageID=0, runInBackground=False*)

This method performs a MICRESS restart which starts/continues a simulation until the time defined in 'tstep'. The external MICRESS application will always run in the background and only one instance of MICRESS will be running at a time. The internal 'locIndex' is an index in the list of different values of the RVE location property.

**Note** This example will skip the steering of MICRESS itself. The parameter 'stageID' and 'runInBackground' are ignored in the current implementation

**Parameters** **tstep** (*Mupif.TimeStep*) – simulation time step including target time

**terminate** ()

This method is used to clean up an interface, e.g. terminating running external MICRESS instances.

**Note** does nothing in this example and also still experimental

**wait** ()

This blocking method waits for an external process of MICRESS to end.

**Note** not in this example; there is no external MICRESS instance to wait for.

## micress.micressConfig module

MICRESS interface configuration file

This file provides necessary information about the MICRESS installation which should be steered by the interface.

**note** In this example, MICRESS will not be run directly. All result data is pre-calculated

## Module contents

MICRESS application interface to be used by MuPIF (Multi-Physics Integration Framework)