

The logo graphic consists of two blue triangles. One triangle is on the left, pointing right, and the other is on the right, pointing left. They meet at a central point, creating a horizontal line that divides the space. The word 'materialise' is positioned to the left of this central point.

materialise

innovators you can count on

API Change Log

Materialise Mimics Medical 25.0

1 Introduction

This document contains a list of changes to the Mimics API. You can use it to find which new APIs have been added to Mimics and to determine what API improvements will affect the calls of the API in your scripts. The overview below describes how the API has been extended and modified between version 24.0 and version 25.0. More detailed API documentation, including examples and tutorials, can be found in the other sections of the Scripting Guide.

2 New API added in this version

2.1. List of API

mimics

- mimics.DataContainerBaseIterator
- mimics.Osteotomy
- mimics.get_registry_path

mimics.file

- mimics.file.ImageDataStorageType

mimics.logging

- mimics.logging.flush
- mimics.logging.get_logs_path
- mimics.logging.get_system_log_file_path
- mimics.logging.get_user_log_file_path

mimics.segment

- mimics.segment.SegmentVesselParameters
- mimics.segment.SegmentVesselPreset
- mimics.segment.activate_calculate_polylines_from_part
- mimics.segment.activate_ct_heart
- mimics.segment.activate_local_threshold
- mimics.segment.activate_multiple_slice_edit
- mimics.segment.activate_segment_vessel
- mimics.segment.activate_smart_brush
- mimics.segment.calculate_ct_heart_right_automatic
- mimics.segment.create_segment_vessel_parameters
- mimics.segment.local_threshold
- mimics.segment.segment_vessel

2.2. Comments

API has been extended to support the new functionality that is available via the Mimics user interface. For more additional information please visit [Help – Scripting Guide](#) and search for the required API.

mimics

- `mimics.DataContainerBaseIterator`

A utility object – iterator for `mimics.DataContainerBase`

- `mimics.Osteotomy`

A new object type that is used only in products created on top of Mimics. It cannot be created via Mimics user interface or Python API.

- `mimics.get_registry_path`

A new API is added that allows getting the root path to the settings of the current instance of Mimics in the Windows Registry.

mimics.file

- `mimics.file.ImageDataStorageType`

New enumeration that specifies possible image data storage types. It is used in `mimics.ImageData.storage_type` property and as a “`image_storage_type`” parameter in `mimics.file.save_project` API

mimics.logging

- `mimics.logging.flush`

A new API is added that allows flushing the log MEMORY buffers to the files on the hard disk. This ensures that all logging records, including those which are currently buffered in memory, will be available if you open one of the log files while a session is still open.

- `mimics.logging.get_logs_path`
- `mimics.logging.get_system_log_file_path`
- `mimics.logging.get_user_log_file_path`

Also, new APIs are added to new `mimics.logging` module to allow you to get paths to the folder containing the log files, to the system log file, and the user log file for the current session of Mimics correspondingly.

mimics.segment

- `mimics.segment.SegmentVesselParameters`
- `mimics.segment.SegmentVesselPreset`
- `mimics.segment.create_segment_vessel_parameters`
- `mimics.segment.segment_vessel`

In short, new functions and objects are added to `mimics.segment` mode that allows you to work with the new Segment Vessel feature. Refer to the Help - User Guide for extra information on this functionality. For detailed examples of these API usages please refer to Help – Scripting Guide.

- `mimics.segment.activate_calculate_polylines_from_part`
- `mimics.segment.activate_ct_heart`
- `mimics.segment.activate_local_threshold`
- `mimics.segment.activate_multiple_slice_edit`
- `mimics.segment.activate_segment_vessel`
- `mimics.segment.activate_smart_brush`

Also, new APIs are added that allow you to activate tools in the Mimics user interface: Calculate Polylines from Part, CT Heart, Local Threshold, Multiple Slice Edit, Segment Vessel, and Smart Brush correspondingly.

- `mimics.segment.calculate_ct_heart_right_automatic`

In addition to the existing APIs that allow automatic segmentation of all heart chambers and only the left heart chambers, a new API was added that automatically segments masks of only the right heart chambers based on CT data.

- `mimics.segment.local_threshold`

New API is added to the Segment module to support new Local Threshold functionality: it adds and removes voxels of a mask and its immediate surroundings by adjusting the threshold of the mask and the size of the surrounding region of the mask

3 Improved API from version 24.0

3.1. List of API

mimics

```
mimics.DataContainerBase
mimics.ImageData
mimics.ImagesContainer
mimics.Part
mimics.disable_update_gui
mimics.disabled_gui_update
mimics.enable_update_gui
mimics.is_update_gui_enabled
mimics.move_object
mimics.rotate_object_around_views
```

mimics.transform_objects
mimics.update_gui

mimics.analyze

mimics.analyze.Point
mimics.analyze.Spline

mimics.file

mimics.file.export_mesh_to_abaqus
mimics.file.export_mesh_to_abaqus_as_single_output
mimics.file.import_mimics_project
mimics.file.open_project
mimics.file.save_project

mimics.measure

mimics.measure.Angle

mimics.segment

mimics.segment.activate_edit_mask
mimics.segment.activate_interpolate3d
mimics.segment.calculate_ct_heart_automatic
mimics.segment.calculate_ct_heart_left_automatic

mimics.view

mimics.view.View

3.2. Comments

mimics

- mimics.DataContainerBase

Iterator object (`__iter__`) `mimics.DataContainerBaseIterator` was added for `mimics.DataContainerBase`

- mimics.ImageData

New properties are added to `mimics.ImageData` object: “`is_dicom_data_modified`” that allows checking whether DICOM data was modified and “`storage_type`” that returns a new enumeration `mimics.file.ImageDataStorageType` that specifies the storage type of the corresponding image data.

Also, a new method is added “`get_dicom_frame_indices`” that returns a list of indices of all (selected and not selected) image frames of an image stack created from a multi-frame DICOM file.

- mimics.ImagesContainer

A new method is added “`reset_sorting`” allowing to restore the original order of images within each study

- `mimics.Part`

The signature of the “color_scheme” property of the `mimics.Part` object was extended by adding the vertex color sensitivity to the color scheme. Refer to the Help - User Guide for extra information on the Vertex Color feature.

- `mimics.disable_update_gui`
- `mimics.disabled_gui_update`
- `mimics.enable_update_gui`
- `mimics.is_update_gui_enabled`
- `mimics.update_gui`

Examples were added to the Scripting Guide illustrating how to use these APIs.

- `mimics.move_object`
- `mimics.rotate_object_around_views`
- `mimics.transform_objects`

The functionality of these APIs was extended by adding the support of `mimics.Osteotomy` object to the corresponding operation on the object.

mimics.analyze

- `mimics.analyze.Point`

New properties are added to `mimics.ImageData` object. “follow_position_of” property specifies the object of a certain type. When the object specified in this property will change its position, the point will follow it and the point position will be changed accordingly. Another new property “allow_change_follow_position” switches ON/OFF the re-link behavior of the point to the object specified in 'follow_position_of property' when the point is edited manually on viewports.

- `mimics.analyze.Spline`

A new property is added “indexes_visible” which allows enabling/disabling visualization of the numbers shown next to the control points of the spline.

mimics.file

- `mimics.file.export_mesh_to_abaqus`

An exception is thrown in case the user tries to export a mesh with an incompatible Abaqus name.

- `mimics.file.export_mesh_to_abaqus_as_single_output`

An exception is thrown in case the user tries to export a mesh with an incompatible Abaqus name. Also, if the “create_assembly” parameter is True Point objects, that are visible and are either assigned to the active image stack or not assigned to any image stack, will be exported along with the volumetric meshes automatically.

- `mimics.file.import_mimics_project`

The signature of the APIs was extended by adding the optional boolean parameter “`ignore_objects_with_existing_guids`”. If True then only 'new' objects will be imported. A 'new' object is defined as the following: its GUID is not present among that of the objects of the currently loaded project

- `mimics.file.open_project`

The signature of the APIs was extended by adding the optional boolean parameter “`read_only_mode`”. If True, the file will be opened in read-only mode which means that, by default, if the project is changed it can only be Saved-As under another file name. Opening a file in a read-only mode can be slightly faster in a concurrent multi-user environment when the file is already locked by another user.

- `mimics.file.save_project`

The signature of the APIs was extended by adding the optional parameter “`image_storage_type`” of the `mimics.file.ImageDataStorageType` type that allows specifying the type of storage of the target file.

mimics.measure

- `mimics.measure.Angle`

A new property is added “`reflex_angle`” which specifies if the angle is acute (always less than 180 degrees) or reflex (always more than 180 degrees).

mimics.segment

- `mimics.segment.activate_edit_mask`

The signature of the API was changed: a new parameter “`enable_clipping`” was added that allows enabling/disabling clipping in the activated Edit Masks tool. Also, the order of “`edit_type`” and “`edit_mode`” parameters was swapped to be consistent with the order of the corresponding user interface controls in the Edit Masks tool.

In 24.0:

```
mimics.segment.activate_edit_mask(mask=mimics.data.masks[0], edit_type="Lasso", edit_mode="Draw")
```

In 25.0:

```
mimics.segment.activate_edit_mask(mask=mimics.data.masks[0], edit_mode="Draw", edit_type="Lasso", enable_clipping=True)
```

- `mimics.segment.activate_interpolate3d`

The signature of the APIs was extended: now it is possible to specify all the parameters in the activated 3D Interpolate tool.

- `mimics.segment.calculate_ct_heart_automatic`

The signature of these APIs was changed. Parameter “`bounding_box`” was removed since now the specification of the ROI is done automatically by the algorithm. A new optional boolean parameter “`segment_myocardium`” was added specifying whether to segment myocardium or not.

- `mimics.segment.calculate_ct_heart_left_automatic`

The signature of the API was simplified: parameter “`bounding_box`” was removed since now the specification of the ROI is done automatically by the algorithm.

mimics.view

- `mimics.view.View`

The support of Panoramic and CrossSectional view types is added.

4 Removed API

4.1. List of API

No APIs were removed.