
What's New in MeVisLab 2.2?

What's New in MeVisLab 2.2?

Table of Contents

1. What's new in MeVisLab 2.2?	5
1.1. MATE as separate program	5
1.2. Integrated Python Debugger	5
1.3. MeVisLab Help Files	6
1.4. Profiling Improvements	7
1.5. IDE Improvements	8
1.6. Ray Casting	8
1.7. Contributions by Fraunhofer MEVIS	9

List of Figures

1.1. Python Debugger	6
1.2. MATE Help Editor	7
1.3. Profiling Call Graph	7
1.4. Profiling Function Details	8
1.5. Snippets View	8
1.6. Endoscopic First Hit Ray Casting	9

Chapter 1. What's new in MeVisLab 2.2?

For a complete list of changes, please have a look at the Release Notes.

The following changes are discussed in this document:

- [Section 1.1, "MATE as separate program"](#)
- [Section 1.2, "Integrated Python Debugger"](#)
- [Section 1.3, "MeVisLab Help Files"](#)
- [Section 1.4, "Profiling Improvements"](#)
- [Section 1.5, "IDE Improvements"](#)
- [Section 1.6, "Ray Casting"](#)
- [Section 1.7, "Contributions by Fraunhofer MEVIS"](#)

1.1. MATE as separate program

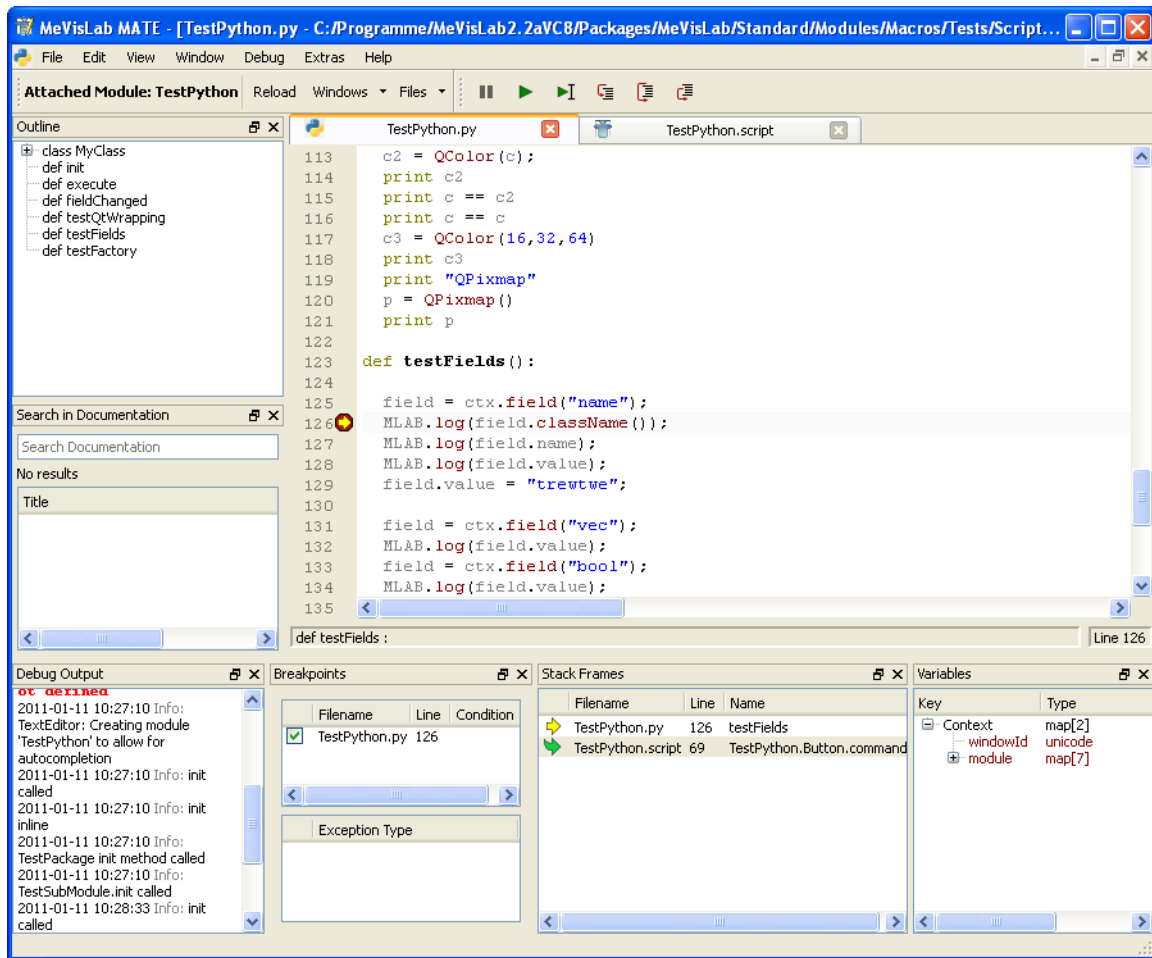
- MATE (MeVisLab's text editor) is now a standalone program.
- It communicates with running MeVisLab instances (if available) for auto-completion and other needs.
- MATE keeps running, regardless of a MeVisLab restart or crash.



1.2. Integrated Python Debugger

- MATE now contains a GUI front-end for MeVisLab's integrated Python debugger.
- The debugger supports:
 - Breakpoints in Python code and on exceptions
 - Conditional breakpoints
 - Introspection of the stack frames including all local variables
 - Breaking into running Python code
- Tedious setup of external debugger is no longer needed.

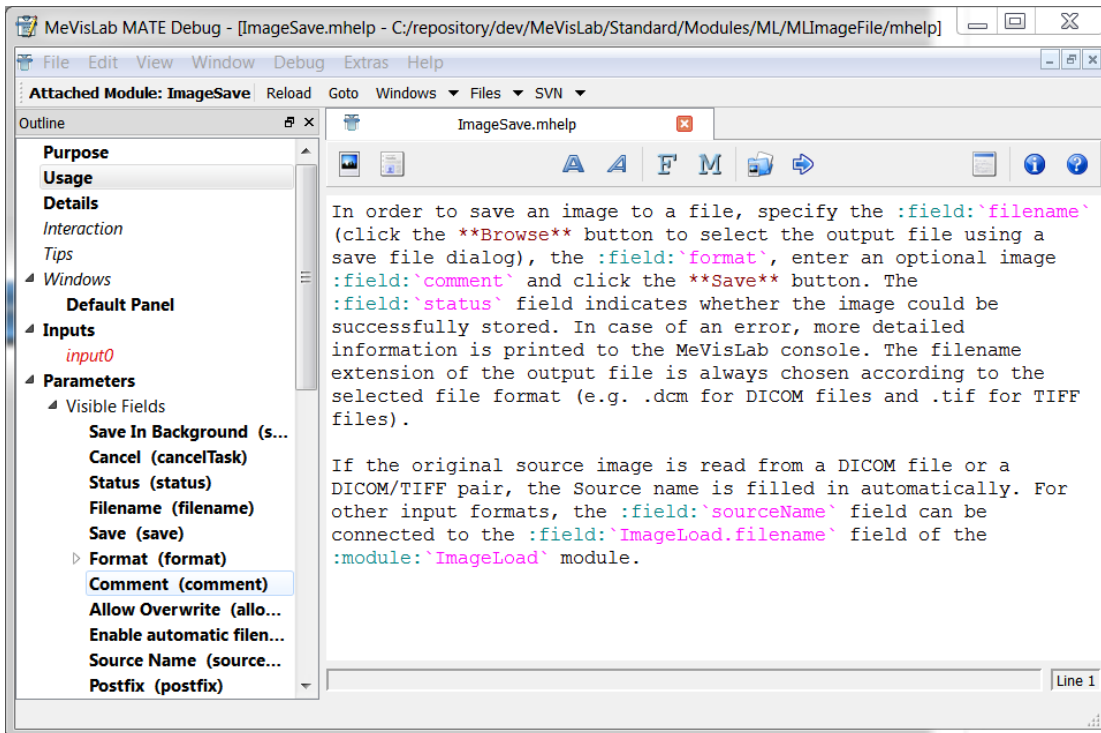
Figure 1.1. Python Debugger



1.3. MeVisLab Help Files

- MeVisLab now uses a new file format for module help pages (*.mhelp).
- MATE offers a comfortable editor for editing module help pages.
- Simple markup based on Sphinx/ReStructured Text.
- Offers easy crosslinking between modules/fields and other content.
- Simple to integrate images and to auto-generate panel screenshots.

Figure 1.2. MATE Help Editor



1.4. Profiling Improvements

- Profiling now supports function profiling and offers a call graph view.
- The call graph now includes: FieldListeners, MDL commands, Python calls and MeVisLab API/Qt calls.
- Function details view shows calling and called functions overview.

Figure 1.3. Profiling Call Graph

Function	Type	Elap	Elap	Self	
renderers.subVolumeStartX	Field Notification	1.6...	82...	21...	...
renderers.subVolumeStartY	Field Notification	1.6...	82...	21...	...
renderers.subVolumeStartZ	Field Notification	1.6...	82...	21...	...
View3D.script(282).FieldListener.command	MDL Command	1.4...	1.4...	19...	...
triggerViewAll	Python Function	1.2...	1.2...	92...	...
MLABBoolField.getProperty(value)	Python Qt Function	6.5...	6.5...	6.5...	...
MLABMacroModule.field(QString)	Python Qt Function	30....	10....	30....	...
MLABTriggerField.touch()	Python Qt Function	28...	14...	27...	...
clip.resetSurround	Field Notification	0.1	0.1	0.1	...

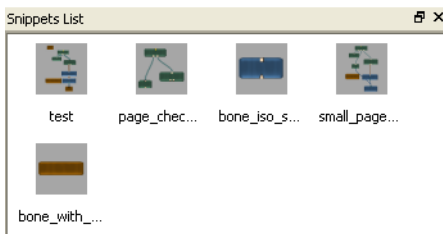
Figure 1.4. Profiling Function Details

Called By	Elapsed Time	Elapsed Time	Callee	Elapsed Time	Elapsed Time	Self Time	Self Time Per
TestPyth...	182.3640 ms	182.3640 m	MLABM...	45.7123 µs	11.4281 µs	45.7123 µs	11.4281 µs
			QRect.se...	2.9256 µs	2.9256 µs	2.9256 µs	2.9256 µs
			QRect.se...	6.9482 µs	3.4741 µs	6.9482 µs	3.4741 µs
			QRect.n...	6.9483 µs	6.9483 µs	6.9483 µs	6.9483 µs
			MLABW...	4.6484 ms	4.6484 ms	4.6484 ms	4.6484 ms
			QSize.____	6.9482 µs	3.4741 µs	6.9482 µs	3.4741 µs
			QSize.se...	2.9256 µs	2.9256 µs	2.9256 µs	2.9256 µs

1.5. IDE Improvements

- New snippets view allows to store/reuse commonly used network snippets.
- New menu options and toolbar that offer the alignment of modules.
- Image inputs/outputs can be colored according to state in the network view for diagnostic purposes (See Preferences/Appearance).
- Preview feature for internal networks - press Space when no module is selected.

Figure 1.5. Snippets View

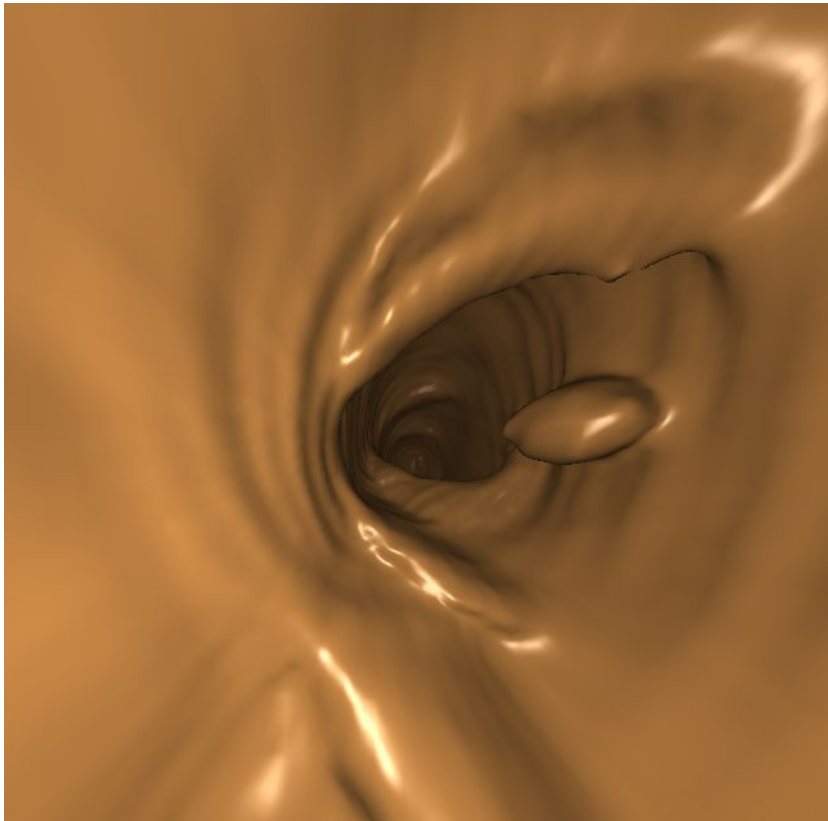


1.6. Ray Casting

- A full-featured GPU Ray Caster has been added to the GVR Volume Renderer.
- The ray caster is fully extensible via the shader pipeline.
- Features:
 - Single pass ray caster with empty-space skipping.
 - First hit ray caster for iso-surface and endoscopic rendering (including hit point refinement).
 - Integration of opaque OpenGL geometry.
 - Supports all features of the GVR, including multiple volumes, tag volume, per tag shading etc.
 - New render modes: MIDA, AverageIP, ClosestVessel.
 - Fully extensible via SoGVRShaderFunction.

- New GVR modules:
 - SoGVRRayCastSettings
 - SoGVRFirstHitRayCastSettings
 - SoGVRFirstHitAmbientOcclusion
 - SoGVRPointLight
 - SoGVRLitSphereShading
 - SoGVRShaderParameterDirection
 - SoGVRShaderParameterPlane
 - SoGVRShaderParameterPosition

Figure 1.6. Endoscopic First Hit Ray Casting



1.7. Contributions by Fraunhofer MEVIS

- ITK/VTK version update and new modules.
- MeVisLab OS X has been further integrated with Mac OS X and optimized for OS X 10.6 Snow Leopard.
- Substantial improvements to the MeVisLab OsiriX Bridge to support multiple MeVisLab targets and to improve the usability of the OsiriX plugin
- New 2D-Viewer 'View2DTouch'. By using gestures, an image dataset may be explored from the trackpad of modern notebooks or by using an external trackpad like the Apple Magic Trackpad.