
MeVisLab Release Notes

Table of Contents

Release Notes	3
Version 4.0.2 Stable Release (2024-09)	3
New Features	3
Fixes	3
Version 4.0.1 Stable Release (2024-07)	4
New Features	4
Fixes	4
Version 4.0 (2024-05)	4
Deprecations	4
Fixes and Enhancements (MeVisLab)	7
Modules and interfaces provided by Fraunhofer MEVIS	10
Releases before 4.0	11

Release Notes

This document lists the most relevant changes, additions, and fixes provided by the latest MeVisLab releases.

Version 4.0.2 Stable Release (2024-09)

New Features

- Qt has been updated from version 6.6.3 to 6.7.2. Other assorted third-party packages have also been updated.

Fixes

- Fixed pixel spacing and patient orientation for single images imported with `DicomImport`, when the corresponding tags are contained in a `SharedFunctionalGroupSequence` tag.
- Fixed that `DicomImport` would sometimes swallow the image data of files when imported together with other files (in the same series) not containing image data.
- Fixed: `DicomImport` would set incorrect image min/max values for files where these values were not explicitly given but had to be derived from the number of bits used.
- Fixed: `DicomImport` did not correctly handle PALETTE COLOR images. This led to garbled images or crashes on accessing the output image.
- Fixed an occasional crash of remote modules using the `RemotePanelRendering` module for remote UI.
- Clean-up and fixes of license identifiers displayed for some third-party libraries.
- Fixed that the context help function in the MATE editor (e.g., for scripting functions) wouldn't jump to the relevant position in the help file.
- Fixed: The `OutputInspector` would show the first inspected output twice if the view is in a floating window (instead of docked to the main window).
- Fixed: Editing network notes did not allow to add links by shift-clicking GUI elements anymore.
- Fixed: Some mouse interactions on a network didn't work on displays with a configured scaling other than 100%.
- Fixed: The ML Module wizard had created invalid code.
- Fixed some duplicated WEM help entries when using the **Help** → **Search in MeVisLab** function.
- Removed some help sections pertaining to the discontinued macOS support.
- ADK/ApplicationBuilder: Removed non-functional macOS settings from the Standalone Application wizard dialog.
- ADK/ApplicationBuilder: Fixed stand-alone applications not working correctly if they use the asyncio event loop, but do not (even indirectly) import the Qt Python bindings.

Version 4.0.1 Stable Release (2024-07)

New Features

- Various small updates for third-party libraries. A notable change is boost, which was updated to version 1.85.0 (from 1.84).
- The `PythonPip` module will now try to determine a SPDX compatible license identifier for newly installed Python packages when generating the `.mlinfo` entry for the **About...** box.

Fixes

- Windows: The SDK installer didn't install the Visual C runtime installer, so it failed if the runtime wasn't installed by some other application first.
- Fixed: `DicomRescale` could crash if operating on a DICOM file with invalid tags.
- Fixed various other crashes caused by loading invalid or corrupt DICOM or `.mlimage` files.
- Fixed loading of DICOM tags which have more than one allowed VR type.
- Updated the MESA OpenGL driver to fix a crash triggered by the `SoPostEffectAntiAliasing` module when using software rendering on Windows.
- Fixed a crash that could occur in the `ListView` control when adding more columns (via scripting) to items containing rich text.
- Fixed a crash in secure mode testing when opening panels with stored relative positions.
- Fixed involuntary pinning of wheel events in the `RemotePanelRendering` module.
- Fixed: The `BoundingBox` module didn't crop the C/T/U dimension information to the output dimensions (unlike, e.g., the `SubImage` module).
- MATE: Fixed hardly readable text in workspace file finder.
- Small documentation fixes.

Version 4.0 (2024-05)

Deprecations

We mainly used this release to remove old interfaces and modules that were deprecated, or to fix interfaces that had severe drawbacks.

Modules

The following deprecated modules have been removed from MeVisLab (you could instantiate them in previous versions if you knew their exact name, or if they were used in old networks).

- `SoAsciiText`
- `CSOBulgeEditor`, `CSOFreehandProcessor`, `CSOIsoProcessor`, `CSOLiveWireProcessor`,
`CSOModifyProcessor`, `CSOPrimitiveProcessor`, `CSOTransformationProcessor`,
`SoView2DCSOEditor`

- SoFixedFunctionShader
- SoAngleLines, SoAngleToObjects, SoCake, SoDistanceLine, SoMainAxis, SoMinimalDistance, SoRuler, SoShapeToPointSet, SoThresholdToPointSet,
- SoScreenSpaceAmbientOcclusion
- ImportDialog, OISettings, OpenImage, SaveImage
- DicomBrowser, EatDicomImport, DicomService
- Negation
- AngleLines2D3D, DistanceLine2D3D, SimpleImageStatistics
- ObjMgrEventClient (and all ObjectManager related modules)
- LiveWireMacro
- SwapViewer
- BaseBypassOp
- ConnectedComponents
- ContourManager, Draw2D, LiveWire, RegionToContour
- DistanceTransform
- LoG
- KernelFilter
- WEMClip
- MovieCreator

Python

- The following MeVisLab-specific Python packages were deprecated and have been removed:
 - `ObjMgr` (as the associated modules have been removed, too)
 - `matrix4` (use `linear.matrix4`)
- The following MeVisLab-specific Python functions and methods were deprecated and have been removed:
 - `linear.vec3.copy` (use `vec3` constructor instead)
 - `linear.toVec3` (use `vec3` constructor instead)

QtWebKit

The old QtWebKit-based web control has been removed in favor of the (still supported) QtWebEngine based one. This new web control supports modern web standards and is based on the Chromium engine (used by the Chrome browser), but has one important drawback: Since the HTML/JavaScript engine runs in a separate process, communication with MeVisLab needs to be serialized, and viewer controls can't be embedded into an HTML page, at least not directly.

The `TestWebEngineView` module demonstrates the abilities of the `WebEngineView` control.



Note

Technical note: Since the `QWebEngineView` rendering in Qt6 is based on `QtQuick`, we had to switch the `QtQuick` rendering backend to "OpenGL" to enable co-existence with the OpenGL/OpenInventor viewers in MeVisLab.

If you implement your own `QtQuick` based widgets in MeVisLab, we suggest to set the backend to "OpenGL" in your code, too, with the following line:

```
QQuickWindow::setGraphicsApi(QSGRendererInterface::OpenGL);
```

to avoid render behavior changing depending on the module load order.

C++

- Lots of old names of renamed functions and enums have been removed.

There still exists a Python script file to help in the renamings: `Packages/MeVis/BuildSystem/BuildTools/Scripts/replaceDeprecated.py`

You can use this script from MeVisLab with the `MLReplaceDeprecatedAPI` module.

- Some methods were replaced because they returned a `const char*` object, where it was the recipient's duty to free the allocated memory afterwards. In these cases there are methods that take a `std::string` instead.
- Some methods were removed because the task could nowadays as easily be performed with methods from the STL (or Qt).
- The method `ml::Module::calculateOutputImageProperties` now has two arguments. The second one is the `PagedImage` on which to set the image properties (previously obtained with `getOutputImage(outIndex)`).

You might notice that trying to override the method with one argument now causes an error in the compiler. This is intentional, you need to override the method with two arguments.

- `getFieldContainer()` isn't needed anymore, as `ml::Module` is derived from `ml::FieldContainer` anyway.

And so on.

If you are unsure how to replace removed methods, install the 3.7.2 SDK, and compile your code with deprecation warnings enabled, and/or look at the documentation of the methods.

Qt6

Qt has been updated to Qt 6.6. Consequently, some long deprecated classes and methods have been removed. Some of the more prominent ones are:

- `QGLWidget` is no more. Use the newer `QOpenGLWidget`.
- `QRegExp` has been superseded by `QRegularExpression`.
- `QTextCodec`, `QTextEncoder`, and `QTextDecoder` have been removed (not exactly, they still exist in the `QtCore5Compat` package, which we don't include in the SDK). By default, Qt only supports Latin1 and UTF encoding.

- The multimedia framework has undergone some refactoring.

Deprecated Modules

The following modules have been deprecated in MeVisLab.

- MessageBox
- MessageBoxExt
- OverwriteVectorComponent

Fixes and Enhancements (MeVisLab)

IDE

- The default encoding of text files read or written by MeVisLab is now UTF-8, if not specified otherwise.
- The window positions of modules in a network are now stored relative to the screen the IDE main window is on.
- Added a new command line argument `-prefs-variable` to define/override a preferences variable (e.g., 'Locale=en_US').
- Auto-completion for imports in the Python scripting console didn't work for namespace packages (PEP 420).
- When searching for words in an editable text area with the search dialog, the highlighted words were hardly visible.
- MATE: Fixed shortcut **F7** being defined twice in the help editor.
- When converting a local to a global macro, a dummy example network is now created automatically.
- **Ctrl+L** doesn't just switch to the **No Dock Windows** layout now, but toggles between this layout and the previous one.
- When changing the visibility of a connector by removing the hidden tag again, a reload of the module would not update the hidden state.
- Fixed that a field comment from the Interface section is lost if the field is also listed in the Description section.
- MATE: The project workspace search now supports multi-word search.
- The MeVisLabStarter executable (on Windows) doesn't have any special functionality anymore, and is just needed to restart MeVisLab. It sets the `-singleinstance` command line parameter of MeVisLab though, so if you want to re-use a running instance of MeVisLab, you can continue using MeVisLabStarter or set the `-singleinstance` parameter yourself.
- When opening a scripting console for the field of a module (through the context menu), it would not stay on top like other windows.
- Fixed a crash on closing the IDE when the cache size in the GUI is updated one last time.
- Fixed background color of screenshots of panels that included OpenInventor viewers.
- The **About** dialog now has a link to the license text of MeVisLab.

TestCenter

- The `CodeTest` module in the `TestSupport` package got an optional parameter for its `inject` function that allows to ignore output lines matching a regular expression before doing error detection.
- Fixed **Show Enclosing Folder** context menu entry for wrapped `CodeTests`.
- Python coverage ignores code in a `site-packages` directory by default (since it contains third-party code). Also added the ability to specify explicit inclusion of directories through the TestCenter config file ("InclusionExpression").
- Images in test results are restricted in the displayed size (but can be expanded).
- Avoid temporarily opening a console window on test cleanup (only happened on Windows).
- If the TestCenter is running in secure mode on Linux, and a test crashes (with a segmentation fault), the TestCenter will try to give a stack trace (which might include the signal handler functions as top-level entries).

Scripting

- The generation of `ThirdPartyInformation` (About dialog) entries and dependency information for Python packages installed with `PythonPip` got more robust.
- `MeVisLab` now comes with automatically generated `.pyi` files for the scripting wrappers available in `MeVisLab`. These should automatically be found by any Python IDE that uses our Python environment (through `sitecustomize.py`) and help with auto-completion and type checking. If you don't want this, set the environment variable `MLAB_DISABLE_SITECUSTOMIZE` to 1.
- Fixed a crash on destruction of the `SoViewportRegionWrapper`.
- Added the option to not create skeletons in `MLGraphWrapper.createEdge`.
- `MLABScriptProcess` doesn't inherit file handles anymore by default.
- Slightly changed the behavior of `MLAB.setVariable()`, which previously wouldn't change a value that was stored persistently.
- The interface of the class returned by `MLABGraphicsScene.addWebView` has changed somewhat, because there is no native support for adding a web viewer to Qt's `QGraphicsScene` anymore.

Controls

- Fixed layout of `HyperLabel` control with longer texts.
- The `Image` control re-renders SVG images to the displayed size now.
- Keyboard navigation/selection did not work for a `RadioButtonGroup` if it was attached to a field.
- Button titles of `ToolButtonGroups` connected to enum fields were not translated.
- The `PythonTextView` was buggy with regard to the syntax highlighting, which could lead to overlapping line numbers and crashes.

C++

- `ml::CurveList` and `ml::CurveData` are now derived from `ml::RefCountedBase`, which means that these classes should not be deleted explicitly by the user, but implicitly through the use of the

`CurveListPtr` and `CurveDataPtr` classes. This also means that `CurveList` and `CurveData` can't be direct members of another class anymore.

Especially the interface of `ml::CurveList` has been adapted accordingly, with explicit methods for adding and removing of `ml::CurveData` objects.

Application Builder

(for those users with an ADK or Application Builder license)

- When local macros share the same name, only the dependencies of one of these modules would be considered.
- Determining DLL dependencies for stand-alone installers on Windows doesn't require an external tool like `DependencyWalker` anymore (and also became a lot faster).

Windows specific

- MeVisLab 4.0 was built with and requires Visual Studio 2022 now.

Third-Party Libraries

A lot of third-party libraries have been updated in this release.

See the list of available third-party libraries [here](#).

Note that this is not the complete list of third-party components used in MeVisLab, for this you should consult the About dialog of MeVisLab, or use the `ThirdPartyInformation` module.

Some notable updates:

- Qt has been updated to version 6.6.3 (from 5.15).
- The PythonQt scripting wrappers for Qt have been re-created.
- Python has been updated to version 3.11.9 (from 3.9).
- `jemalloc` isn't used by MeVisLab and not included in the SDK anymore.
- MeVisLab now solely uses `libjpeg-turbo` for the JPEG file format. `libjpeg` is no longer included in the SDK.
- CMake target names of some third-party libraries have changed (mostly upper/lower case).
- The Python package `asyncqt` has been replaced by `qasync` to fix problems with `aiohttp`.
- ITK has been updated to version 5.3.0, and the module wrappers have been updated, including newly introduced classes.
- VTK has been updated to version 9.3.0. Almost all module wrappers have been removed, please use the Python bindings provided by VTK instead. Have a look at the example networks of the remaining bridge modules for how interfacing with MeVisLab modules works.
- The Windows version now uses 7-Zip in its installer.

Modules

- `DicomImport`:

- Do not split image series just because they erroneously have slightly differing StudyDate/StudyTime tags.
- Make `DicomImport` (and other modules) more tolerant against certain errors in the DICOM data.
- Do not show corrupt image data for `.dcm/.tif` pairs. Correctly report missing image data (`.dcm/.tif` pairs must be read with `ImageLoad`).
- We added an example `DicomViewer` application module to `MeVisLab`, which provides basic DICOM image viewing capabilities in 2D and 3D.
- The `clear` button in the `WEMInteract` module now clears the drawn contour.
- `WEMModify` updates the face normals now. This attribute is, e.g., used by `WEMInfo` to compute the volume of a WEM patch.
- `SignedEuclideanDistanceTransform`: Fixed wrong output for example network if `minValue` and `maxValue` are set to 0.
- `SoCoordinateSystem` rendered the text shadow in front of the actual text under certain circumstances.
- `CSOMerge` did handle input notifications when auto update was set to off.
- Most WEM modules that use PVLs (primitive value lists): Introduced checks for correct type (per-node, per-edge, or per-face).
- `WEMLoad`: Introduced a new field for auto-loading of WEMs.
- `ConnectedComponentsToImage`: Did set an invalid image extent if no cluster was selected.
- `WEMVascularSystem`: Would hang indefinitely if the graph was completely outside the image extent associated with it (which could happen if you constructed the graph by scripting).

There is also now a field `clipping` which determines if the resulting WEM should be clipped to the associated image extent.

- Added new color modes to `So3DMarkerRenderer` that use vector length/angle and use a LUT as color input.
- `WEMLoad` now can load `.ply` files with one-based indexing (previously it crashed).
- `ImageStatistics`: Fixed that the variance could get slightly negative due to rounding errors.
- Fixed rare crash in `SoCoordinateSystem` if no axis was initialized.
- The `OutputInspector` for images now allows to step through the user dimension with **Ctrl+Left** and **Ctrl+Right**.
- Fixed: The code generated by the Project Wizard for modules derived from `SoView2DExtension` missed an important dependency.

Modules and interfaces provided by Fraunhofer MEVIS

Additions

- Module `DicomPRSave` saves `XMarkerList`, `StylePalette`, and `CSOLists` Base objects as and in DICOM files of Grayscale Softcopy Presentation State IOD. See also `MultiFileVolumeListPROutput`.

- Module `FileEncrypter` encrypts files using a new C++ API `BinaryEncryption`, but the currently implemented encryption method is internal to FME, so this module is only useful if you implement your own.
- Module `SoMLMatrixTransform` wraps `SoMatrixTransform` to enable direct use of ML and numpy matrices (which are transposed relative to the matrices used by `SoMatrixTransform`).

Improvements

- `DirectDicomImport` resolves variables (such as `$(NETWORK)`, `$(TMPDIR)` etc.) in fields **Import Files** and **Import Directories** in the **Configuration** panel.
- Improved robustness of experimental `MultiFileVolumeListPROutput`.
- `LoadYAML` and `ReadYAML` can now deal with Python tuples encoded with `"!!python/tuple"`.
- Macro module `CachedPath` and Python module `file_caching` now support pruning of the synchronization target folder, which is enabled by default.
- Removed the unused/unsupported `NoCache` mode from the modules `ImageCache` and `ImageCacheForNetworkLoops`.
- `ThirdPartyLicenseReport`: now reads 'homepage' as alternative to the 'url' key in .mlinfo files; and some other minor improvements.
- The `fmeTestSupport` Python module now supports `EXPECT_ERRORS()`.

Changes

- The C++ Module base class `XMarkerListProcessor` now allows to specify a number of input images (for modules that combine image and marker data)

Releases before 4.0

For the release notes of older MeVisLab releases see this document.