

# New Features in OpenSim 3.0

OpenSim 3.0 is our biggest upgrade to-date. We've added new features, including live model editing, scripting in Matlab and the OpenSim application, and improved muscle models. We've enhanced some of the core existing OpenSim functionality, including our muscle models and tools for visualizing forces and other motion data. OpenSim 3.0 also includes a host of usability improvements for beginning users to advanced developers. Read more about the new features and improvements below.

- [Live Model Editing in the Graphical User Interface \(GUI\)](#)
- [Scripting from Matlab and the GUI for Batch Processing and Extending OpenSim](#)
- [GUI Usability Improvements](#)
  - [Loading and Saving Models and Motions](#)
  - [Working with the OpenSim Workflow Tools](#)
  - [Creating Plots](#)
  - [Additional GUI Updates](#)
- [Visualizing Models, Motions and More in the GUI](#)
- [New Muscle Models](#)
- [New Probe Component for Making Virtual Measurements](#)
- [API Improvements](#)
- [License Updates for OpenSim](#)
- [Documentation and Examples](#)
- [Bug Fixes](#)

## Live Model Editing in the Graphical User Interface (GUI)

- You can now **edit almost any property of an OpenSim model in the GUI** using the [Property Editor and Outputs List](#). Markers, body masses, muscle max isometric force, and joint locations and other model parameters can be easily adjusted. You no longer have to perform the tedious process of editing, saving, and loading xml files when editing and refining your OpenSim models.
- When editing you can **Undo/Redo** your changes. This includes the [Marker Editor](#) and [Muscle Editor](#), eliminating the old, and sometimes confusing, Backup and Restore mechanism.
- You can **select model objects in the 3D view** window via double click. The selected object will appear in the Property Editor, so you can see its properties and edit them.
- We've added the ability to **save and load marker sets** from a separate XML file, so you can reuse marker sets with different models. Read more in the [Marker Editor](#) page.

## Scripting from Matlab and the GUI for Batch Processing and Extending OpenSim

- Many OpenSim users have a large number of subjects, trials, or designs to study.
  - [Scripting](#) allows you to **automate your research workflows** to accelerate your work or share your workflow with colleagues.
  - You can use **GUI scripting** to access and edit model properties, run tools, create plots, and more.
  - You can use **Matlab scripting** to access and edit model properties, run tools, and call the API visualizer. Plus you can integrate Matlab's native functionality, like its optimizer and plotting abilities.
- More and more users want to **extend the functionality of OpenSim** by accessing its API or application programming interface, but learning C++ and setting up a development environment is a barrier. You can now access much of the OpenSim API using Matlab or GUI scripts.
- We've created documentation and **a host of examples** to demonstrate how to use the new GUI and Matlab scripting features. Read more in the chapter on [Scripting](#).

## GUI Usability Improvements

### Loading and Saving Models and Motions

- You can load model files and motion files in the GUI using **Drag and Drop**.
- You can **Open Recent Files** from the File menu.
- We've added commands to **Close All** and **Save All** models.
- Read more about each of these in [Opening, Closing, and Saving Models](#)

### Working with the OpenSim Workflow Tools

- We've added context sensitive **Help Buttons** in each of the tools. Click on the "Help" button to go directly to the documentation for the tool you are using.
- When you run the [Residual Reduction Algorithm](#), you will see an **RRA Results Summary** in the [Messages Window](#), so you no longer have to sort through RRA's long output list.
- We've added buttons to **Save Settings** and **Load Settings** at the bottom of each of the tools, making this functionality more accessible so you remember to save your settings files.
- Defaults in tool settings are no longer propagated when you save settings files, leading to less clutter and smaller files.

### Creating Plots

- You can **edit figure title and curve names via double click** from the curve list in the plotter
- We've changed the **default figure title** to "Figure 1" for the first figure you create, "Figure 2" for the next, etc. This gives you a more professional-looking plot without having to make any edits to the default output.

- Read the chapter on [Plotting](#) to learn more.

## Additional GUI Updates

- You can **find the installer version** available from the GUI Help -> About window. This allows you to more easily check for compatibility when using Plugins. (See [Creating Plugins](#) for more information about OpenSim plugins.)
- You can **Undock and Redock Windows** in the GUI.

## Visualizing Models, Motions and More in the GUI

- We've made it easier to **associate and sync force and marker data** with loaded motions. Read more in [Associating Data with a Motion](#).
- We've also added the ability to **visualize the results from analyses**, like [Joint Reactions Analysis](#) and [Induced Acceleration Analysis](#).
  - The new features for [Associating Data with a Motion](#) allow you to show forces in different references frames.
  - You can change the scale and color of the visualized vectors, such as ground reaction forces or joint forces.
  - You can show resultants or just components of interest.
  - You have the option to save your setting for later re-use. These settings are in the same format as ExternalLoads, so you can use them interchangeably.
- You can **rotate models** incrementally in the viewer **with j-k hot keys**. Read more in [Navigating the Visualizer Window](#).
- We've implemented **smarter bounding box creation** so models appear in near full size in the viewer. When new models are generated or loaded, there is less offset between models.
- You can **change marker set color** using the [Property Editor and Outputs List](#).
- We've added an OpenSim logo as a small watermark in the viewer and movies.

## New Muscle Models

We have made some big improvements to the existing OpenSim muscle models and added new muscle models. Changes are summarized briefly below. Watch [Matt Millard's Webinar](#) about the new muscle models to learn more.

- We've improved the Thelen2003 muscle model, updating it to better match the original 2003 paper and improving its numerical accuracy.
- OpenSim 3.0 also includes several new muscles (MillardEquilibrium and MillardAcceleration).
- We've revamped the base Muscle class to allow consistent access to key parameters and values (e.g. fiber velocity).
- We've added a new interface to easily modify the force-length and force-velocity curves of a model (in the API).
- To accompany these changes, we've expanded muscle testing and benchmarking to ensure that a muscle model performs correctly.
- Users will now see errors and warnings for non-physiological values (e.g. negative fiber velocity, negative muscle lengths) and other undesired behavior.
- Muscle analysis now has better error handling and modularity.
  - The analysis of a model with multiple muscles will still complete even if some of the muscles in the model fail to equilibrate or produce other errors.
  - Values are reported up to and after the time of failure, to enable better troubleshooting.
  - Results storage at the API level was improved, so that space for results is only allocated as needed.

## New Probe Component for Making Virtual Measurements

- Probes are a new model components that **perform measurements on an OpenSim model** during a simulation.
- These measurements can then be outputted to an external file using a **ProbeReporter** analysis.
- Four Probes are included with the OpenSim 3.0 release, including probes to output **actuator force**, **actuator power**, **joint internal power**, and **stem energy**. You can add any of these existing probes to your OpenSim model.
- Developers and API users can **create new Probes** and combine them with existing Probes to achieve a wide range of functionality.
- Read more in the section on [Probes](#).

## API Improvements

- The **new ModelComponent** interface enables users to more easily add custom model components.
- We've added a **Visualizer**, to visualize OpenSim models and their simulations using the API and thus accessible to user programs.
- We streamlined the **property handling interface** to reduce error incidence and the length and complexity of the code required to add properties.
- OpenSim 3.0 uses Simbody 3.0 as its underlying dynamics engine for **improved performance** and new features. Read more in the [Simbody 3.0 upgrade notes](#).
- We updated the **Coordinate class** to the new property interface, improved its documentation and memory management, and added the property `is_free_to_satisfy_constraints` to allow dependent coordinates to assume any value necessary to satisfy constraints during the assembly of a model.
- We revamped the **Controller class** by converting to new properties, and performing other clean up and interface improvement
- The API now includes more convenient **states reporting**.
- We added **setDescription**, **setAuthors**, **setReferences** methods to Object to enable you to describe your component and identify yourself and your sources, so that users of the components you create can credit your work.
- You can access the Doxygen documentation from the OpenSim GUI.
- We **streamlined the build process** by refining the CMakeLists, adding better documentation, distributing the source code via a zip file, eliminating Xerces, and fixing linux build bugs.
- Read more about the OpenSim API and new class structure in the [OpenSim API Doxygen](#).

## License Updates for OpenSim

OpenSim 3.0 includes updated licenses for the OpenSim application and the API. Read the [License for OpenSim 4.0 and Later](#).

- The OpenSim application now allows **commercial use**, provided you still acknowledge your use of OpenSim
- The OpenSim **API is now fully open source**, following the [Apache 2.0 License](#). We request that you acknowledge your use of the OpenSim API.
- The model files distributed with OpenSim have custom licenses, with most using Creative Commons (CCBY 3.0). Refer to each model for more details.
- Users developing new models, plugins, simulations, etc. retain the right to define their own license.

## Documentation and Examples

- We are now using **wiki-based documentation** for all of our documentation (user guide, developer guide, examples, tutorials, etc.). This documentation is hierarchical and searchable, and makes it easy to constantly add new information and examples. The GUI now links to this online documentation (or a local copy if you're offline). Some key features of our online documentation include:
  - An updated [User's Guide](#) and [Developer's Guide](#) for OpenSim 3.0
  - A central [Examples and Tutorials](#) page
  - A [Modeling and Simulation Best Practices](#) section
  - A [Musculoskeletal Models](#) library, with expanded documentation
  - A section with [Teaching Materials](#)
  - Find out how you can help improve the OpenSim documentation in our [How to Contribute](#) section
- We've added many examples for using the new [Scripting](#) functionality in Matlab and the GUI
- Our **API examples** have also been improved and expanded
  - There is a new plugin example showing you how to create a "BodyDrag" force.
  - We improved the example for creating a custom muscle with fatigue that utilizes the new Muscle interface
  - We've streamlined the API example build process, to help you get started with these examples more quickly.
  - See the [Developer's Guide](#) for more information.

## Bug Fixes

- We corrected an IK bug discovered in the GUI. When specifying IK coordinate tasks, the GUI was using the current value of a model's coordinate when the "value\_type" was set to default\_value. It has been corrected to now track the coordinate's default value (NOT its current value).