Release notes for simulator version 1.0.23
February 25, 2020

Bug Fixes

1. Fix a crash in simviewer when trying to zoom an empty plot.

New Features

1. The model can be saved to a PDF file. This was already part of the code, but for some reason there was no menu item to select. It is in the File menu. Like simviewer, it saves what you see in the model graphic panel. You can save part, or all of a model.

2. A feature was added to the Options Menu to switch the model to monochrome mode. It displays the background as white and all foreground objects as black. The intended use of this is to save a pdf that can be incorporated in a document or to print a model without using all the black ink in the toner cartridge.

If you want to print a complex model on multiple pages, load the pdf file into Adobe reader, select poster in the print setup, and set the scale to a larger value, such as 150%. It will print the model on multiple pages that you can cut and tape together.

3. Minor text change in a few menu items. For example, "Load .snd File" is now "Load .snd Model File." A couple of people have asked, "What is an snd file?"

4. The XY cursor position in simviewer is now always displayed. There is also some text that explains what the numbers at the top of the drawing area are, which are the left time, right time, and width of display in seconds. Changing the time slider will change this. You can also resize the window and see the width in seconds change.

Release notes for simulator version 1.0.22
February 11, 2020

Bug Fixes

1. Deleting rows in the bdt and plot tables in the Launcher Window could produce a corrupted pointer with undefined results. This would only happen when selecting File-New and if there was more than one row in either table.

New Features

1. Add optional different paths to input and output files to simrun. These are only available from the command line. This was added to support the SIMCORE/OSPARC environment. Input and output files need to reside in different folders.

Release notes for simulator version 1.0.21
January 16, 2020

This release fixes some connector line problems in the simbuild graphical editor.

Bug Fixes

1. It breaks things if an incomplete connector is saved as part of a model. There were several places where it was possible to do this. A check and a warning dialog has been added to stop this from happening. You must either delete the incomplete connector or finish connecting it before, for example, saving a file.

Which brings us to:

2. If you deleted an incomplete connector, when you moved the node it was attached to, the program crashed. This has been fixed.

3. Another way to create an incomplete connector is to edit an existing one and move the start or end point to the edge of an object's hot-spot (dotted lines). There is a case that is hard to duplicate where it is possible to move the end a pixel or two outside the hot-spot. When loading a model that has instances of this, the program moves the line, or adds a new one, to ensure it is inside the hot-zone. There was an error in this code that sometimes failed to move the point into the hot-zone. This has been fixed. When this happens, a warning is printed in the info box in simbuild and the model is marked as "needs to be saved." The user should save the corrected model in these cases.

Release notes for simulator version 1.0.20
January 6, 2020

New Features

1. The user's manual has been brought up to date. In the simbuild Help menu, selecting "View Manual" displays this manual. Selecting "View Older Manual" displays the newsned manual. The newsned program assumes that the user will edit text files in situations where using the GUI is awkward, or does not support an operation. The updated user's manual deletes a lot of content that assumed this. Users wishing to edit the .snd, .sim, .ols, and other files in a text editor should consult the older manual.

2. The Linux programs use a plotting program plotmv. This is a twenty year old program that has not been maintained since 1995. The default is gnuplot, which is now the only option. A Windows version of this is available, but it takes more work, such as porting Linux shell scripts to Windows powershell scripts. This work remains to be done.

Bug Fixes

1. When using simbuild's File -> Save Sim File, the corresponding .ols file was not saved. Now it is.

2. Each run of the simulator creates / overwrites several files. A few of these used the launch number as part of the file name. Some did not and would be over-written by the next launch, resulting in unpredictable behavior. Now all of the files generated by each launch have the launch number as part of the file name.

3. Related to this, there was a bug in a Linux script where each launch would over-write the power spectrum file.

4. When using the network connection to send simrun results directly to simviewer, and if simrun completes a simulation faster than simviewer can display it, some of the output results are never displayed. The wave file method does not have this issue. The reason for this is that data written to files are buffered and when the file is closed, the data is written to the file even if the application terminates. Network sockets buffer the data, but any pending data is discarded when the socket is closed or the application terminates. Simrun now sends an End Of Data packet to simviewer at the end of a simulation and then waits for simviewer to send an acknowledgment that is has received this packet. In effect, it tells simrun it has all the results and that it is okay for it to terminate.

This could also occur when running more complicated simulations for a very long time.

Release notes for simulator version 1.0.19
November 25, 2019

New Features

1. In the Launch Window, Create smr File has been added to the left panel. If selected and if there are rows in the bdt table, a .smr file will be created that CED's Spike2 program can read. The bdt file for Launch #0 is always spawn0.bdt. The corresponding .smr file is spawn0.smr. In spike2, if you select View -> File Information, the date the .smr was created and the .snd file used are in the information dialog box. The simulator assigns channel numbers starting at 101. It assigns all of the cell populations first, then the fiber populations. The .smr file also contains the original cell or fiber populations from the model.

On Windows, if the user is viewing the .smr file in Spike2, Windows will not allow the simulator to create a new file because the file is in use. The user should close the file before starting a new run. The simbuild program will display a dialog box warning the user.

2. In support of this, you can sort the bdt table in the Launch Window by clicking on any header. The sort will list cells first, then fibers. The population numbers and member are then sorted in either ascending order or descending order. You can rearrange the rows by hand. If you list Fibers before Cells, the display order in scope and Spike2 will not correspond to the visual order. The program does not automatically sort the table when new items are added.

3. Add a Find menu item and a single choice, Find. Selecting this pops up a list that contains the cell and fiber populations and the Comment text. Clicking on the Find button ensures that the population is visible and selects it. The parameter page for the population is also displayed.

4. Add the current Launch number selected in the Launch Window to ChangeLog files.

5. A command line program that converts existing bdt or edt files to smr files has been included in the package. The Linux version is edt2spike2, the Windows version is edt2spike2.exe. To run it, open a command line window. On Windows you should use the cd command to change to the program's directory. The Windows command is:

```
    edt2spike2 -n c:/path/to/yourfile/somefile.bdt
```

If the bdt file is in the current directory, you do not need to type in the drive or path. The program will create a .smr file that spike2 can read.

Bug Fixes

1. Toggling the Add Axon/Synapse button from the On to Off state did not change the internal state, so the drawing state remained "drawing a connector." Fixed.

2. Fixed a bug where some lung row values in the Launch Window were being cleared out, which caused everything after that row to fail to plot.

Release notes for simulator version 1.0.18
October 24, 2019

1. The vertical time markers were not correctly displayed. For .bdt tick counts, there were twice as many markers as there should be. Selecting a marker space 1000 milliseconds apart displayed markers every 500 milliseconds. The fix takes the tick size into account, so the marker spacing is correct for .edt, .bdt, or custom tick counts.

2. The infinite loop bug fixed in version 1.0.16 occurred in another place in the code. Fixed.

3. Simviewer has a new scaling option. It is called Anti-clip Scale. If this is selected, the Vertical Gain Scaling Factor is used to scale all of the row plots. If the value of a row plot would over or under shoot the row's box, the gain for that row is adjusted to not let this happen. The alternative is to let the plots expand into neighboring boxes, which is the default, or to clip the plot so parts of it are not displayed, which has not been implemented. This and the Auto Scale option are mutually exclusive.

4. The Users Manual has been been updated, but additional work remains to be done.

Release notes for simulator version 1.0.17
October 11, 2019

1. New functionality: All communication between the simbuild, simrun, and simviewer programs are via the network interface. Simbuild still writes spawn[0-14].txt and spawn[0-14].sim files to the local disk, but sends the contents via the network. This is in anticipation of how the SIM-core interfaces between simulation software might be structured. On possible configuration is to use Docker containers. The typical interface between a service (such as simrun) in a container is via a network connection. It is also possible to share files between containers, but it can get complicated with multiple users. But, in the absence of specific information, who knows? In principle , we could run simbuild and simviewer on one computer and run simrun on a more powerful system which could be running under a different operating system. I made a version that did this to see if it works, and it does. The controls required to do this have been hidden in this version. It takes quite a bit of system administration work to configure the systems.

2. Enhancement: When selecting a cell or fiber population, the population number is displayed next to the optional group name on the parameter pages.

3. Bug fix: When selecting a line that has other connector lines underneath it, a popup dialog box is display so you can select one of the connectors. You can click on the column headers to sort the rows. This was broken and how now been fixed.

4. Bug fix: If you selected a connector and change the type, this was not flagged as a change and you would not be prompted to save or undo the change if you selected another connector. This has been fixed. This had the side-effect that the color of the connector was shown (incorrectly) as the color of the new type. The (correct) color of the current type was not displayed until you selected the connector a second time.

5. Bug fix: A line of code in the build_network function was included inside a debug block. The effect is that it would not execute unless debugging was turned on. The effect of this is dependent on the model. In some cases, it would never execute even when correctly include. Since it allocate memory and returned a pointer, I would expect simrun to crash or produce random simulation results. All copies of version 1.0.16 should be deleted from your systems.

Release notes for simulator version 1.0.16
September 26, 2019

NOTE1: Starting with this usfsim release, all release notes are contained in a single file. Versions will be in most-recent-first order.

NOTE2: The last formal release was version 1.0.11. There were several internal releases that were used for troubleshooting some program defects specific to the Windows version.

1. An electric stimulation fiber type has been added (abbreviated e-stim.) To use this, add a fiber population to the model, then select the new fiber item in the drawing. The Fiber Parameters page now has two options, Normal and Electric Stimulation. Some of the parameters are the same as the Normal type. The parameters specific the e-stim are:

> Deterministic or Fuzzy: If deterministic, e-stim is applied at exactly the same time as determined by the frequency. If fuzzy, e-stim is applied at a random time centered on the deterministic time.
>
> Frequency: How often the stim is applied, in Hz.
>
> Fuzzy Frequency Width: If fuzzy, the stim will be randomly applied sometime in the millisecond interval centered around the deterministic time.

Note that there is not a Fibers In Population parameter. The e-stim fiber type has a hard-wired population of one.

2. Units have been added to many of the spinner controls.

3. Several parameters that were set in tick counts are now set as milliseconds or seconds. The tick counts represent time values, but the actual value will vary if you change the tick interval. If you want to have a start time at one second and select a tick value of 0.5 ms, the start tick count is (doing math in head) 2000 ticks. If you change the tick value to 0.1, 2000 ticks is no longer one second. The presentation of these parameters is now in floating point seconds, so, for example, if you want a start time of 150 milliseconds, you would enter 0.150 seconds. These are converted back into tick values when the simulation is run. The program knows that older model files have the values in tick counts, and converts them to time values using the tick value and the step interval. The file will be updated to the current format when it is saved.

4. You could not view the ChangeLog.txt file if a model has not been loaded. Now you can, but this is only the case for a file named ChangeLog.txt. If you are using the model file name as a basename for a change log file, you still have to load the model first. Of course, these are text files, so can be viewed in any text editor.

5. By default, the launcher window is always on top of the simbuild window. If you want to look at the simbuild window, you have to move it out of the way. There is now an option in the Options menu to toggle this, so you can click on the simbuild window and have it be on top of the launcher window. The windows are independent in this mode, so you have to bring the Launcher window to the foreground by hand. When in the always-on-top mode, bringing the simbuild window to the foreground also brings the launcher window to the foreground. Clicking on the Open Launcher button in the simbuild window will always bring the launcher window to the foreground.

6. The program remembers the last parameters for a cell or fiber population. When you add a new cell or fiber population, these will be the defaults. Previously, the default was always a fixed set of values.

7. When adding an item to the bdt table in the launcher window, a random member of the population is generated. If the number has already been selected, a different random number is generated. If all of the members of the population have already been used, the program hung in an infinite loop. Fixed.

8. The internal tables used when creating and editing the launcher bdt and simviewer tables were not cleared out before copying the current values to the tables. This resulted in deleted rows still being sent to the simrun program. Fixed.

9. Fiber populations have Time to Begin and Time to End parameters. If you do not want the firing to end, you had to enter a value that was larger than the simulation time. If you increased the simulation time, you may exceed the End value. Now, if you select an end value of -1, either explicitly or by using the down arrow to pick the next value less than zero, the program will display "Never". Regardless of the value of the simulation run time, the value for the end time sent to the simrun program is guaranteed to be larger.

10. Several issues that were Windows specific have been fixed. Many of these had to do with using the wrong character in path names. Linux expects a '/' character, Windows expects a '\' character.

11. It is possible when moving the connector end points in a drawing to have the coordinates outside of the dotted circle or rectangle that represents the object's "hot zone." When loading a model where this has occurred, the program failed to make the final connection to the object and the connector was in a state where it could not be selected. If this occurs, the program now moves the endpoint to be inside of the "hot zone." It prints a warning message and marks the drawing as changed. The user will be prompted to save the file if they want to run a simulation, load a new file, or close the program. The user should save the adjusted drawing so this does not happen again.

12. Some additional popup tooltip help has been added to some of the controls to make the program more self-documenting.

13. I am working on updating the user's manual. A lot of the GUI-specific information is no longer accurate and there are conceptual changes as well. The current work-in-progress has been included in the package. It is *usfsim.pdf.* For Linux systems it is located in /usr/local/share/doc/usfsim. For Windows systems, it is located in the directory where the .zip file was unzipped.

14. If network errors are encountered in the Windows version, popup dialog boxes are displayed to communicate possible sources of the errors.

Release notes for simulator version 1.0.13
September 6, 2019

This is an in-progress version that has new software that may assist in diagnosing problems running on Windows. There are some pop-up dialog boxes and text that is printed in the text output controls in simbuild and simviewer.

Screenshots of the dialog box and a copy and past operation of the text in the text controls can be emailed to dshuman@health.usf.edu.

The new electric stim fiber type is not completely implemented, so leave that alone.

Release notes for simulator version 1.0.11
August 14, 2019

In addition to some bug fixes and cosmetic changes, this release has four new features.

1. GROUPS

- There is a new menu, Groups.
- A set of cell and fiber nodes can be selected and then assigned to a group.
- Some (or all) of a model can be in a group.
- A node can only be in one group at a time.
- A node can be removed from a group.
- Connectors belong to a group only if both endpoints are in the group, so the user does not need to select connectors.
- Groups do not nest, that is, a group cannot contain another group.
- There is a page in the Help dialog box that has details of how to manage groups.

2. EXPORT AND IMPORT

- Once you create a group, you can export the group to a .sndlib file.
- The parameters, shared connectors, synapse types, and geometric information is saved in the file.
- The population and synapse numbers are adjusted to start at one.
- You can import one or more .sndlib files into a model.
- Population and synapse numbers are assigned using the next unused numbers.
- The intended use is to create functional subsystems that can be reused in new models.
- Once a group has been imported, the user must make connections between
- it and other nodes in the model.

You can select a subset of a model and export it without assigning it to a group. It makes managing the imported nodes more difficult, so in general, use groups for exporting.

I discovered something during testing that should be mentioned. I loaded a simple model, exported the entire model as a group, then imported it into an empty model, expecting the simulation output to be identical. It was not.

The reason why is that each fiber and cell population has a unique random number seed that is internal to the program that the user cannot change. This should not be confused with the random number seeds that the user can change. If you create model one with fiber pop1 and cell pop1 and create a set of parameters and run a simulation, you will get a result that will be the same each time. However, if you create a new model two by adding in cell pop1 and cell pop2, fiber pop1 and fiber pop2, and then delete cell pop1 and fiber pop1 and create an identical model using fiber pop2 and cell pop2, you will always get the same result for each run, but the results will be slightly different from model one.

The colors of the connectors are likely to be different because the synapse type is what determines the color. The synapse type number in the original model may be different when importing a group. The functionality remains the same, but it may be a bit confusing visually. If the imported types are identical to existing types, the user can change the type.

3. WAVE FILES ARE OPTIONAL

- This version uses network sockets for communication between simbuild, simrun, and simviewer.
- The newsned program wrote the simulation results to a series of wave files.
- This can create thousands of files.
- By default, simrun now sends the contents of the wave files directly to simviewer.
- There is an option in the simbuild Launcher window to select direct communication or wave files.
- If you do not want to view the results again, used direct communication.
- If you want to review the results at a later time, use wave files.

You can run the simviewer program in stand-alone mode to view wave files. By default, when run with no options, it runs in wave file mode using launch number 0. You can use the window controls to navigate around the file system and pick which launch number you want to view.

4. CHANGELOG NAMES

By default all changes are written to a file named ChangeLog.txt. There is an item in the simbuild Options menu to add the basename of the current .snd file to the file name. If you work with several .snd files in the same directory, you can get a change log file specific to each model.