Simbody 2.2 release notes and change log (since 2.1, Aug 2010)

Michael Sherman, 14 May 2011

The primary change in Simbody 2.2 is a drastically simplified build, distribution, and installation process, including separating out Simbody instead of bundling it with other software in the "SimTKcore" project. Changes include

- Combining several separate projects into a single Simbody source tree for a one-step build process. Simbody now includes what were formerly simtkcommon, cpodes, and simmath projects; those projects are now obsolete.
- Eliminating dependency on VTK. Simbody has a completely new Visualizer that depends only on OpenGL and GLUT. (More below)
- Eliminated the simbody_aux library (was required for VTK use).
- Use native lapack and blas when available (Mac and Linux).
- Provide a source distribution "zip" file, duplicating exactly the source structure that was used to produce the prebuilt binaries. Step-by-step build from source instructions are now provided.
- Binary packages are now built with zip rather than installers, making the installation process transparent.

How to get help

Other than the documentation, the best source of help is the Simbody "help" forum. Go to the Simbody home page https://simtk.org/home/simbody. Select "Documents" tab for documentation, including detailed API documentation (see below). Or, select "Advanced" tab, "Public Forums", then click on "help". Note that we are no longer using the SimTKcore project forum for Simbody support.

There is also a new community Wiki accessible from the Simbody home page. If you don't see what you need there, please consider adding it once you resolve your problem to make it easier on the next person with the same problem. You can also attempt to get others to add to the Wiki by filing a feature request or posting to the forum.

Our help forum is friendly – don't be shy about posting. You won't be ridiculed if it turns out your question was already clearly answered on page 43,271 of the documentation :-).

API Documentation

Many of the notes below say "see API documentation" meaning the Doxygen-generated detailed API reference information extracted from the code. That documentation is in html so must be viewed from a brower. It is located in your Simbody installation's doc subdirectory, with the starting file doc/SimbodyAPI.html. It is also posted online on the Simbody project's Documents page, go to https://simtk.org/home/simbody, Documents tab, select "Simbody 2.2 API

Reference", or use this link:
https://simtk.org/api_docs/simbody/api_docs22/Simbody/html/index.html.

Many improvements have been made to the Doxygen documentation in this release, and the layout has been simplified. Some general topics are available under the "Modules" tab. If you know the name of a class you want more information about, type the name into the search box and click on one of the resulting links.

New Visualizer

Simbody is a numerical library independent of any visualization system or GUI. However, it is very useful to have default visualization capability. Previously Simbody shipped with a visualization capability based on VTK (VTKVisualizer). This was a misuse of VTK and provided minimal capability, ugly graphics, and mediocre performance. It also greatly increased the complexity of building and installing Simbody as well as dramatically increasing its size. We have now replaced that with a completely new, lightweight Visualizer that is much faster, produces better images including ground and shadows, supports interactive and real time capability, and generates snapshots and movies. The Visualizer is almost completely backwards compatible with the old one so most user programs will continue to work as is, or at most require very minor changes.

The new Visualizer operates from a separate executable that resides in the Simbody installation's bin directory (named VisualizerGUI[.exe]). That executable has some dependencies on the local machine's graphics capability, including OpenGL and GLUT. The Simbody library contains a new class Visualizer, and some related support classes, that do not introduce any such dependencies. The library-side class, if used, will attempt to launch the VisualizerGUI executable and communicate with it via standard interprocess communication.

The best way to get a sense of the new Visualizer's capabilities is to build and run some of the examples whose source is provided in the installation's examples/simbody directory. Looking through the code for those may also be useful. For documentation of the library-side interface, read through the Visualizer class API documentation here:
https://simtk.org/api_docs/simbody/api_docs22/Simbody/html/classSimTK_1_1Visualizer.html
(This is also in the installation's doc directory; aim a browser at doc/SimbodyAPI.html then search for Visualizer.)

Misc. usage notes

- Main header to include is now just "Simbody.h" (was SimTKsimbody.h which still works).
- No need to call System methods updDefaultSubsystem(); instead the System accepts calls like addEventHandler() directly now.
- The library name is still SimTKsimbody. If you are using shared objects (dynamically linked libraries) on Linux or Mac you should

need only to specify -lSimTKsimbody; all the other library dependencies will be discovered automatically.
- On Linux, the Simbody libraries now have a dependency on the standard "librt" real time library. If you are building with static libraries (not common), you will have to add "-lrt" explicitly to the link line.
- Visual Studio 8 (2005) is no longer actively supported. You should still be able to build from source using that compiler, though. See the "how to build from source on Windows" document.
- Visual Studio Express users will need to build from source also.

New features

The main new features are the simplified structure and new Visualizer described above. There are some smaller additions also:

- An assortment of high precision easy-to-use timing functions was added to support the Visualizer's real time capability. See https://simtk.org/api_docs/simbody/api_docs22/Simbody/html/group__TimingFunctions.html for more information (or, got to API documentation, select "Modules" and look for "Timing Functions").
- Functions for exact and approximate calculation of complete elliptic integrals were added to support Hertz-based elliptical contact: https://simtk.org/api_docs/simbody/api_docs22/Simbody/html/group__EllipticIntegralsGroup.html
- Previously-declared but unimplemented method calcGt() in SimbodyMatterSubsystem is now implemented – this returns the transpose of the constraint matrix G.
- Added MobilizedBody method calcH_FMCol() to get the local mobilizer hinge matrix H_FM (in addition to the already-available body-to-body one calcHCol() returning H_PB_G).
- ParallelWorkQueue class to facilitate use of parallel threads to process a single queue of tasks. This is used to speed up movie generation in the VisualizerGUI but is generally useful. See API documentation: https://simtk.org/api_docs/simbody/api_docs22/Simbody/html/classSimTK_1_1ParallelWorkQueue.html
- CoordinateDirection class adds +/- sense to CoordinateAxis class, see API documentation.
- You can now add visualization "hints" to a System object: use a CoordinateDirection to indicate "up" (default is +Y), and say whether the default ground/sky background is appropriate for viewing this System. See API documentation for System class.
- System class now forwards addEventHandler(), etc. methods to its internal subsystem, so it is no longer necessary to use updDefaultSubsystem(). This makes for nicer examples and cleaner code in general.
- Several improvements were made to the Xml classes per user requests: can now set indent string for pretty-printing; added an Xml::Document subclass; documents and nodes can be cloned; nodes

can be extracted and moved around; some memory management bugs were fixed; better Doxygen documentation.

Experimental features

- Ellipsoid/half plane contact has been implemented and is useful for foot-ground contact in place of contact spheres. Ellipsoids are a much better approximation of foot shape. See Rattleback example or EllipsoidContact test case or post to the forum.
- ContactDetail class has been implemented and can be used to extract detailed information about elastic foundation contact patches, such as pressure per element. See ExampleContactPlayground for code using this to display pressure and velocity profiles of contact patch.
- Added Measure::Result, a built-in measure convenient for allocating cache memory that will be automatically invalidated upon state changes. This is now used by OpenSim to collect up control signals (controls) from an assortment of controllers. See the API documentation for more information: https://simtk.org/api_docs/simbody/api_docs22/Simbody/html/classSimTK_1_1Measure___1_1Result.html
- There is now a Simbody Wiki accessible from the Simbody home page. We would like community contributions. If you run into a problem and solve it, or if you make a cool example that could be useful for others, please post it there.

Bug fixes and minor improvements

- Removed unnecessary "restriction" layer from State class to reduce overhead.
- State::invalidateAll() now requires non-const access to the State being invalidated since this can wipe out Model-stage state variables.
- Continued purging the API of getNBlah() methods in favor of getNumBlah() for consistency.
- Numerous minor changes to support clean compilation with gcc 4.4 (e.g. Ubuntu 10) in 32 and 64 bit modes.
- CMake script now disables several optimization for gcc 4.4.3 to avoid compiler bugs.
- Mac only: Fixed bug in Pathname class that prevented it from finding the correct current executable's directory on Mac
- PolygonalMesh now has shallow (reference counted) copy semantics to save space
- Windows only: DLLs now go in the bin directory in keeping with Windows convention (they were in the lib directory which is the Mac/Linux convention). Now only the bin directory needs to be in the PATH.
- Windows and Mac: added missing emulations for Posix time-handling methods clock_gettime(), nanosleep(), usleep() (already available on Linux). But see "timing functions" under new features above for better functions.

- Cleaned up some namespace intrusions. Now very close to the promised land of introducing only symbols in the SimTK namespace or beginning with "SimTK_". Please report as a bug if you find that any other symbols sneak into your namespace when you use Simbody.
- Much cleanup of Doxygen documentation. Most of the internal classes that were inadvertently included have been eliminated. If you see any more garbage in the Doxygen documentation, please let us know (file a bug, post, or email).
- Added some missing "const" specifiers on some get() methods.
- Added Mobod and MobodIndex as official abbreviations for MobilizedBody and MobilizedBodyIndex.
- Renamed Gyration class to UnitInertia, which is much clearer.
- Templatized MassProperties class by precision to match other related classes.
- Many doxygen documentation improvements throughout the code.
- State object now allows time to be less than zero.
- MobilizedBody::getHCol()was incorrectly taking UIndex (global index); should have been MobilizerUIndex (local 0-5 index).