
VSim Installation and Release Notes

Release 8.2.1

Tech-X Corporation

Jan 30, 2018

CONTENTS

1	VSim Introduction	1
2	VSim System Requirements	3
3	VSim Installation Instructions	5
4	VSim Documentation	11
5	Release Notes	13

VSIM INTRODUCTION

Welcome to VSim, the relativistic, arbitrary dimensional, hybrid plasma and beam simulation code. Below is a detailed description of the software parts installed.

1.1 Included Software

VSim consists of two major components:

- Vorpal - the VSim computational engine (“vorpall” for parallel, “vorpalsr” for serial)
- VSimComposer - the graphical user interface

Upon completing the installation process (described in *VSim Installation Instructions*), both components will be installed on your computer along with Python and MPI.

1.1.1 VSimComposer

VSimComposer is a graphical user interface for

- Creating and editing VSim input files
- Executing VSim
- Analyzing VSim generated data
- Visualizing VSim generated data
- Viewing the documentation.

The VSimComposer editor and validator have built-in functions and graphical components that help you to create well-formed input files. Example input files, ranging in complexity from beginning to advanced, are included with VSimComposer. New VSim users can use these examples as templates. Advanced VSim users can use VSimComposer to validate the syntax of their own input files, whether their files have been created using VSimComposer or by using a text editor.

The VSimComposer Run pane invokes the VSim engine with user definable settings for number of steps, number of data dumps, and restart file, if any. It also allows selection of serial versus parallel VSim.

VSimComposer now allows selection of analysis programs, either supplied with VSim or user written.

The visualization in VSimComposer is provided by the graphical analysis tool *VisIt*. *VisIt* is embedded within VSimComposer. Data generated by VSim or by analysis programs automatically appears in the Visualization pane.

All documentation can be seen from within VSimComposer, fully cross-referenced.

1.1.2 VSIm Computational engine

The VSIm computational engine runs both as a serial (vorpalsr) and parallel (vorpals) application for multi-processor / multi-core systems that support MPI. VSIm now comes in the specialized VSIm packages. VSIm is embedded within VSImComposer.

1.1.3 Python

Python is an open-source, interpreted scripting language managed by the Python Software Foundation. For more information about Python, see: <http://www.python.org/>

VSIm uses Python to process input files, allowing users to set up simulations with math functions, variable substitutions, and macros.

VSIm uses its own embedded version of the Python interpreter to pre-process input files and execute any Python code in an input file. Python is embedded within VSImComposer.

1.1.4 MPI

The Message Passing Interface (MPI) is an application programming interface (API) for communicating between processes that execute in parallel. There are many implementations. The Linux and Mac versions come with the OpenMPI (<http://www.open-mpi.org/>) implementation of MPI. The Windows versions come with the Microsoft MPI implementation (<http://www.microsoft.com/>). The appropriate MPI implementation is embedded within VSImComposer.

1.2 More Information

More information about VSIm can be found at the [VSIm Product Website](#). Send questions about installing or running VSIm to Tech-X Customer Support at support@txcorp.com. For technical questions about the physics, users may wish to email the VSIm discussion list ([vorpals-users List](#)). Extensive assistance in the use of VSIm or simulation in general is available from Tech-X Professional Services. Please contact Tech-X directly for sales, consulting, and other questions at sales@txcorp.com.

VSIM SYSTEM REQUIREMENTS

VSIm runs on 64-bit Windows, Linux, and Mac and has installation procedures that users will be familiar with on their respective operating system. Some of the systems supported are:

- Windows 7, 8, 10, Server 2008 R2, Server 2012
- Linux distributions with glibc 2.11.3 or later (verified on Fedora/RedHat/CentOS, SUSE, Ubuntu)
- Compatibility version will support Linux distributions with glibc 2.5 or later (verified on CentOS and Ubuntu)
- Mac OS X Yosemite, El Capitan, Sierra, and High Sierra
- Cray XC30

Note: The version of glibc can be found with the command: `ldd -version`

The VSIm serial engine (vorpalsr) is available for running on single processor workstations. The VSIm parallel engine (vorpal) is provided for multi-core systems that support the Message Passing Interface (MPI). There is support for running VSIm simulations on Linux clusters with common job schedulers as well as running with the “Windows Clustering” server technology.

VSIm is particularly scalable, and has been developed to solve the most challenging computational electromagnetics problems of our time. Some example calculations, particularly for plasma acceleration, require a supercomputer to run adequately in 3D, though VSIm desktop users may explore how the files work by running them in 2D. Some microwave devices examples such as our HelixTWT power run could require 12GB RAM or more in your computer to run, however it is possible to set up an input file at lower resolution if you have more limited resources.

The VSIm 8 Windows installer is around 350MB, and requires around 1.2GB of disk space to install. The Linux installer is slightly over 1GB due to the inclusion of additional system packages, and unpacks into around 2.7GB. Please ensure you have enough additional space to run your simulations.

The visualization in VSIm use OpenGL and requires optimal graphics drivers. The standard Linux distributions may not come with drivers written by the graphics-card manufacturer, which are necessary for full hardware acceleration. You should download and install the latest driver for your graphics card from your graphics card vendor’s website. In the case of an NVidia graphics card, you can get the latest driver by going to NVidia’s website, selecting the Download Drivers link, and then selecting the Linux Display Drivers link.

The application has a minimum size that does not work well below the resolution of 1280x800 pixels or XVGA. Smaller resolutions will start to lose parts of the user interface of the application.

VSIm has GPU computing capability based on the NVidia CUDA Toolkit 7.5. This is supported only on 64-bit Linux and 64-bit Windows. One does not need a CUDA-enabled graphics card to use VSIm unless the GPU capability is invoked, but if a simulation is configured to run on the GPU, then the graphics card must have CUDA capability and a driver that is compatible with version 7.5 of the toolkit.

Note: To use the GPU capability of VSIM, you will need a CUDA-enabled GPU card with a driver that is compatible with the CUDA Toolkit version 7.5. Downloading the developer toolkit is not needed, just the driver, but if you do download the CUDA Toolkit (see <https://developer.nvidia.com/cuda-downloads>, then it will guarantee your driver is compatible.

For large parallel simulations running across nodes in high performance cluster, there are some requirements that can worked out easily with system administrators and depend on the details of the cluster configurations. In general, we don't recommend running VSIM on AFS file systems.

Note: The Andrew File System (AFS) is not recommended to run VSIM in parallel. The distributed AFS system is optimized for location-transparency over a wide area network rather than the low-latency conditions necessary for high performance in cluster storage. File systems such as Lustre or the General Parallel File System (GPFS) perform better when running VSIM.

For installation instructions see: *[VSIM Installation Instructions](#)*

VSIM INSTALLATION INSTRUCTIONS

These are instructions on installing the VSim product. Please see the *VSim System Requirements* first to make sure your system meets the requirements.

3.1 Windows VSim Software Installation

The VSim distribution package for 64bit Windows is a self-extracting executable installer. Invoke the installer by double clicking on it. The default installation path is:

```
C:\Program Files\Tech-X (Win64)\VSim 8.2
```

To open the VSim software, go to the Start Menu, click on the Tech-X folder, click on VSim 8.2, then click on VSimComposer. See *Dialogs for Windows installation*.

3.2 Windows VSim Software Installation Using a Windows Network Location

These instructions outline how to install VSim to a shared network location. This method will allow users from multiple machines to access and run this installation of VSim on their local machines.

To start, you must have a shared network location that you intend to install to. To do this, locally navigate to the location you wish to install to. Right click the target folder and select properties. A window will appear. Select the “Sharing” tab near the top of the window. Click Share and set the permissions appropriately.

Now that this folder is accessible, install to the folder by following the Windows Installation instructions, using the shared location as your installation location. This can be done from the machine where VSim is being installed, or from another machine with access to the shared folder.

In order to run VSim from a network location, it is best to map the shared network location to a network drive on any machine where VSim may be run. To do this, simply right click the shared network drive on the machine where VSim will be run and select Map Network Drive. At this time you will be prompted to select a drive letter and change settings. Change the settings as desired and press Finish. The user should use this path rather than the Network path when starting VSim. When running VSim from a shared network location, we suggest using the VSim executable directly by navigating to:

```
Contents\bin\VSimComposer.exe
```

For convenience, a shortcut can be made on the machine in which VSim is run. Ensure that the shortcut uses the mapped network drive.

To ensure all users have access to the license, the license should go in:

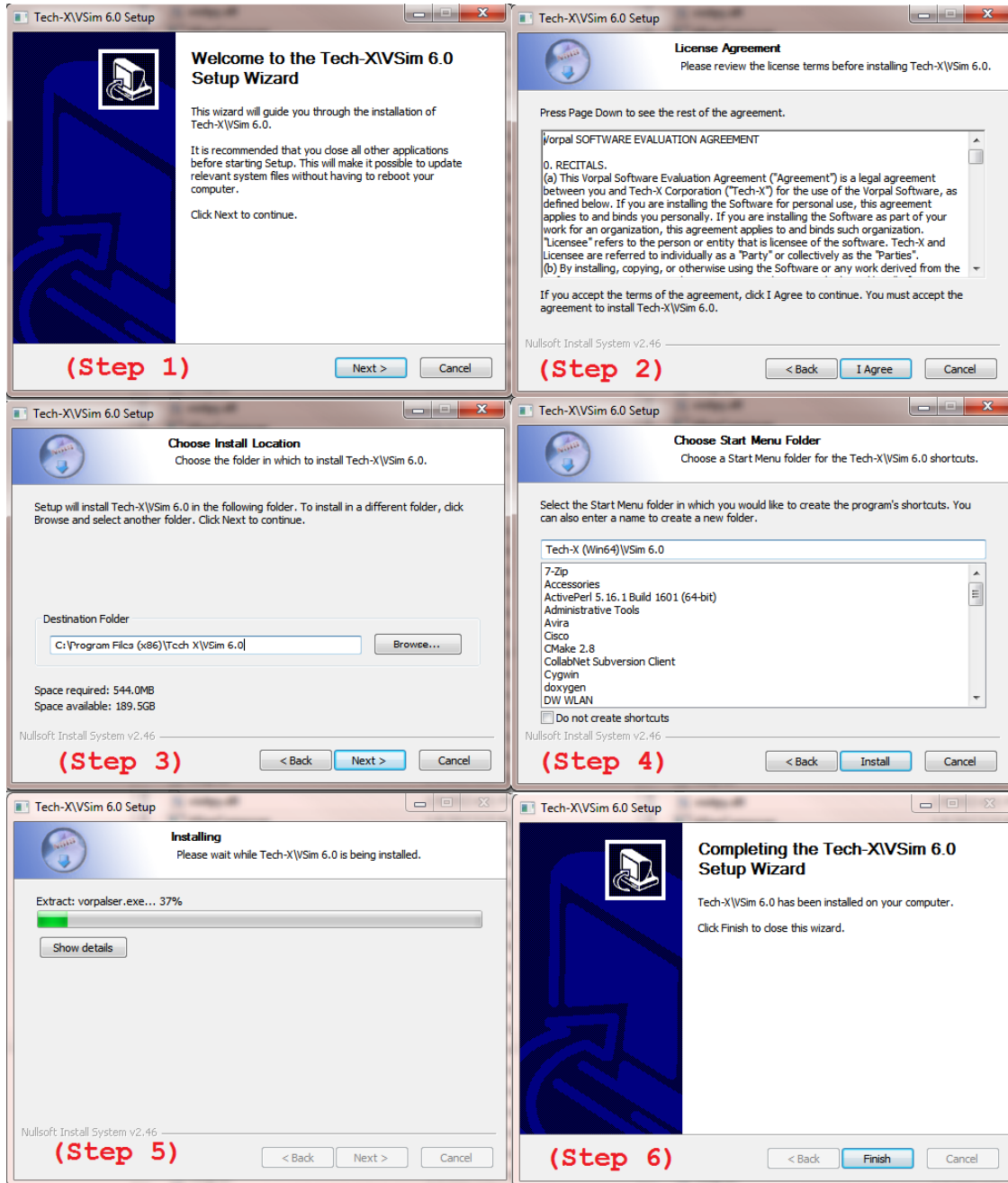


Fig. 3.1: Dialogs for Windows installation.

```
Contents\engine\bin\
```

inside the installation of VSImComposer.

3.3 Linux VSIm Software Installation

The VSIm distribution package for Linux is a gzipped tarball. Unpack the gzipped tarball into the directory in which you wish to install VSIm. A typical location would be

```
/usr/local/VSIm-8.2
```

The unzip and untar command is

```
$ cd /usr/local
$ tar -zxvf VSIm-8.2-Linux64.tar.gz
```

Or, if you are installing the compatibility version then the file would be “VSIm-8.2-Linux64-glibc2.5.tar.gz. Once, untar, then the user interface is started with the command:

```
$ cd VSIm-8.2-Linux64
$ ./VSImComposer.sh
```

If you plan to run the simulation engine or any other executable from the command-line then you will need to source the startup script:

```
$ source VSIm-8.2-Linux64/VSImComposer.sh
<execute engine, analyzers, etc.>
```

See [user-guide-running-vorpal-from-the-command-line](#) for more instructions on command-line operations.

3.4 Mac OS X VSIm Software Installation

The VSIm distribution package for Mac OS X is a .dmg installer. Invoke the installer by double clicking on it. Drag the VSIm-8.2 folder into your Applications folder (visible in the installer window). From the Application folder, double click on the VSImComposer icon in the VSIm-8.2 folder. See *Dialog for the Mac installer.*. This default installation path is:

```
/Applications/VSIm-8.2
```

3.4.1 MacOS Sierra Users:

First Method

- Make sure you have deleted ALL instances of the version of VSIm that you are about to install, including desktop icons.
- Unmount any VSIm installer disk image. DO NOT CLICK ON IT!
- In a terminal window, run:

```
sudo spctl --master-disable
```

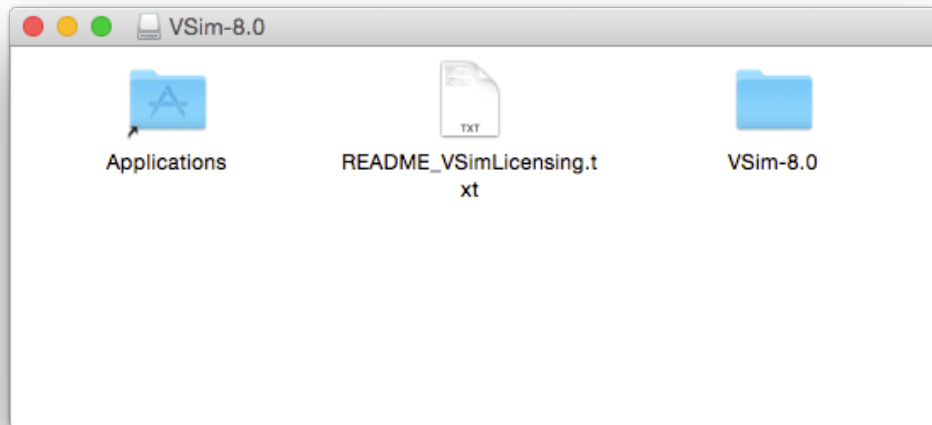


Fig. 3.2: Dialog for the Mac installer.

- Run the disk image in your download folder to install VSIm.
- A window should open that shows your Application folder and a VSIm-8.2 folder.
- Drag the VSIm-8.2 folder into the Applications folder, then run VSImComposer from inside VSIm-8.2 folder that is inside the Applications folder.
- Open VSIm-8.2 from the Applications folder and add a license.
- AFTER adding the license, in a terminal, run:

```
sudo spctl --master-enable
```

Second Method

- Make sure you have deleted ALL instances of the version of VSIm that you are about to install, including desktop icons.
- Unmount any VSIm installer disk image.
- Double click on the VSIm installer disk image to mount it.
- A window should open up that shows your Application folder and a VSIm-8.2 folder.
- Drag the VSIm-8.2 folder into the Applications folder.
- In a terminal window, run:

```
sudo spctl --add /Applications/VSIm-8.2/VSImComposer.app
```

- Run VSImComposer from inside that VSIm folder.
- You will now be able to add a VSIm license.

If you open VSIm and get a message saying “License File Not Found”, make sure that the path for your license starts with:

```
/Applications/VSIM-8.2/...
```

: _vsim-documentation:

VSIM DOCUMENTATION

In addition to this PDF version of the VSim documentation, all of the documentation is accessible from within the VSimComposer interface, as well as online at the Tech-X web site, [VSimDocumentation](#)

4.1 VSim Quick Start

vsim-quick-start provides numerous tutorials for both beginning and advanced VSim users.

4.2 VSim In Depth

vsim-in-depth contains comprehensive VSim documentation, including directions for running VSim from the command line support resources. When you are ready to create your own simulation, consult this document for in-depth information about VSim features.

4.3 VSim Reference

vsim-reference is a quick-reference manual for VSim users to look up specific VSim features and code block syntax for use in editing a VSim input file.

4.4 VSim Installation

VSim Installation Instructions guides the user through the installation process for VSim. Release notes are also provided in this document.

RELEASE NOTES

The release notes describes new features for the VSim computational engine (Vorpal) and the VSimComposer graphical user interface. Deprecated features and bug fixes are also noted within the release notes.

5.1 New and Updated VSim 8.2 Features

The following features are new or have been updated for the 8.2.0 release.

5.1.1 VSim Computational Engine (Vorpal)

- Added a new Monte-Carlo elastic collision algorithm with threshold energy functionality
- Removed assumption in elastic collisions of first incoming particle being light compared with second incoming particle with the result being energy conservation in the collisions of particles of similar masses.
- Fixed memory leaks with some histories on long runs

5.1.2 VSim Graphical User Interface (VSimComposer)

- Removed gstreamer-0.10 requirement on Linux

5.1.3 VSim Macros and Translator

- Capitalization fix in embcs.mac

5.1.4 Other

- Improved general robustness of license management
- Made cluster and node-locked license tokenfile behavior consistent between Vorpal and VSimComposer

5.2 Known Issues in VSim 8.2

- Remote VSimComposer does not work
- Windows Cluster support was lost with upgrade to MPI on windows needed for certain simulations.
- EM Cylindrical not implemented for visual setup

- Occasionally, for parallel runs, the stl reader can spuriously set some regions outside of objects to being inside. When this happens, it is manifest in the visualization of the corresponding geometry field in Data Overview. In all cases observed so far, this has been fixed by offsetting the grid a small amount, changing the number of cells, and/or changing the number of parallel processes.

5.3 New and Updated VSim 8.1 Features

The following features are new or have been updated for the 8.1 release.

5.3.1 VSim Computational Engine (Vorpal)

- Scattering processes added for antimony (Sb), including electron-electron scattering
- Added transport and emission from antimony (Sb) using the density of states
- Added model to specify percentage of specular and diffusive emissions
- Added Species kind “bandSpeciesES” to the VSimSD license
- Fixed issues with several commandline arguments including -h and -help
- Improved warnings when improper values are given for simulation parameters
- Fixed issue with line dumping without a name
- Fixed issue where all 3 components of deltaAtBreaks were not checked to determine if the grid is uniform
- Fixed energy conservation for binary elastic Monte Carlo collisions
- Improved error recovery and reporting for the Filter Diagonalization analyzer

5.3.2 VSim Examples

These examples now have Visual Setup:

- A Loop Antenna created from a coaxial cable (VSimEM) : See Example
- Horn Antenna (VSimEM) : See Example
- Patch Antenna Far Field (VSimEM) : See Example
- Rectangular Waveguide (VSimEM) : See Example
- Electron Gun (VSimMD) : See Example
- Gyrotron Mode (VSimMD) : See Example
- Helix Traveling Wave Tube 1: Dispersion (VSimMD) : See Example
- Helix Traveling Wave Tube 2: Impedance and Attenuation (VSimMD) : See Example
- Helix Traveling Wave Tube 3: Power Run (VSimMD) : See Example
- 2D Magnetron (VSimMD) : See Example
- Multipacting Resonance in Waveguide (VSimMD) : See Example
- Multistage Collector (VSimMD) : See Example
- 3D Stripline Multipacting (VSimMD) : See Example

Other example features include:

- Updated examples to take advantage of new macro functionality
- Added the numerical Cerenkov filter option to EM examples
- Improved Photonic Crystal examples

5.3.3 VSim Macros and Translator

- New macro to allow numerical Cerenkov filters in Visual Setup as a parameter of the field solver
- New macro, stfuncs.mac, to handle built-in Space-Time functions including Python file import and feedback
- Added support to set localVelocity parameter off of shape emitters
- Updated the reflecting particle boundary macros to handle rails and corners
- Added history to let user set pseudoPotential by coordinates in addition to via grid index
- Added support for secondary emitters to test particles
- Added support to simplify creation of rectangular and coaxial waveguides
- Added pipe primitive
- Fixed issue with HalfWave Dipole Antenna failing on GPU
- Fixed several issues dealing with handling of cylindrical coordinates
- Improved performance of filters macro
- Improved embcs.mac and VSimEm.mac for a faster implement of MAL boundaries
- Improved particles.mac (support for particle species loader with repeat loading)
- Removed unused macros, remove unneeded parameters, and cleaned up referenced inside macros
- Fixed several issues with histories.mac for logging of ptclQty and speciesAbsPtclData
- Fixed issue with parallel restarts on Win10 with CAD geometries
- Fixed issue with specifying name of external magnetic fields
- Better handling of global variables in macros

5.3.4 VSim Analyzers

- Improvements to data analysis executables. : See Analysis Scripts
- New analyzer, exportSpecies.py, that exports a particle species to a text file or sequence of text files
- New analyzer, getFieldComponentsOnPlane.py, to get field components on a plane
- Updated the computePtclNumDensity.py analyzer to perform spatial averaging
- Improved addPtclComponentKEeV.py, fixing bug involving mismatched components after multiple runs, adding check on restart, executing dump files in numerical order, better specifying of KEeV, KEeVx, KEeVy, or KEeVz, all in one script

5.3.5 VSim Graphical User Interface (VSimComposer)

- Improvements to tree-based/visual setup : See Visual-setup Simulations
- Fixed issue in visual setup current sources changing field specified from E-Field to a proper J-Field
- Better handling of Space-Time functions in Visual Setup tree
- Added support for Field Scaling Electrons
- Improved Constructive Solid Geometry tessellation and meshing for geometry primitives
- Improved speed of Visual Setup rendering
- Added support for geometries described by Python functions
- Improved expression handling in Visual Setup tree
- Fixed issues with certain plots incorrectly displaying in Visualization Tab
- Fixed automatic scrolling issues in Run and Analyze Tabs
- Added support for scaling and translation of STL files
- Added support for all macro changes above in Visual Setup tree
- Fixes issues with resizing of the grid
- Improved usability and robustness of Visual Setup tree interface
- Improved wildcard searching in Documentation/Help
- Improved searching with omitting terms with minus sign in Documentation/Help
- Reduced the verbosity of search results in Documentation/Help
- Added instructions on searching in Documentation/Help
- Improved navigation in sidebar contents in Documentation/Help
- Fixed issue with equation rendering speed in Documentation/Help.
- Improved appearance of overall Documentation/Help formatting
- Improved documentation of Visual Setup
- Improved operation under node-locked license

5.3.6 Other

- Over 50 use issues fixed.

5.4 Known Issues in VSim 8.1

- Remote VSimComposer does not work
- Windows Cluster support was lost with upgrade to MPI on windows needed for certain simulations.
- EM Cylindrical not implemented for visual setup
- Occasionally, for parallel runs, the stl reader can spuriously set some regions outside of objects to being inside. When this happens, it is manifest in the visualization of the corresponding geometry field in Data Overview. In all cases observed so far, this has been fixed by offsetting the grid a small amount, changing the number of cells, and/or changing the number of parallel processes.

5.5 New and Updated VSim 8.0 Features

The following features are new or have been updated for the 8.0 release.

5.5.1 VSim Computational Engine (Vorpai)

- New package: VSim for Semiconductor Devices for modeling electron transport in diamond or Gallium Arsenide : See [VSim for Semiconductor Devices](#)
- Fast dielectric algorithm can take multiple dielectrics assigned to multiple shapes : See [multiDielectricUpdater](#)
- Windows Cluster Support with graceful shutdown of job : See [Windows HPC Cluster](#)
- Automatic detection of GPU acceleration : More info on [GPU Computing](#)
- Ability to use cross-section data files from a user definable location : See [SIM_DATA_PATH](#)
- 17+ new simulation object kinds (including Species, Particle Sources/Sinks, Emission Models, Histories, Field Updaters, Grids, Grid Boundaries, and Domains)
- Improved reliability of fast electrostatic solves on linux

5.5.2 VSim Examples

- Charge Carrier Dynamics in Diamond (VSimSD) : See [Example](#)
- Electron Transport in GaAs (VSimSD) : See [Example](#)
- MESFET Device (VSimSD) : See [Example](#)
- Metal Schottky Contact and Transport in Diamond (VSimSD) : See [Example](#)
- Transport and Emission in Diamond (VSimSD) : See [Example](#)
- Electron Emission in GaAs (VSimSD) : See [Example](#)
- Langmuir Probe (VSimPD) : See [Example](#)
- Turner Plasma Discharge, Case 2 (VSimPD) : See [Example](#)
- Electron Gun (VSimMD) : See [Example](#)
- Half-Wave Dipole in Free Space (VSimEM) : See [Example](#)
- Linear Phased Scanning Array (VSimEM) : See [Example](#)
- Gaussian Laser Beam and Photonic Crystal Cavity (VSimEM) : See [Example](#)
- Dipole Source and Photonic Crystal Cavity (VSimEM) : See [Example](#)
- Scattering off Multiple Objects (VSimEM) : See [Example](#)
- Drude-Lorentz Metal-Insulator-Metal Waveguide (VSimEM) : See [Example](#)
- Spherical Lens (VSimEM) : See [Example](#)
- Halfwave Antenna (VSimBase) : See [Example](#)
- Cylindrical Capacitor (VSimBase) : See [Example](#)

5.5.3 VSIm Macros

- Macros standardized and simplified : See Version 8 Macro Infrastructure
- Ability to use macros from a user definable location. : See TXPP_PATH

5.5.4 VSIm Analyzers

- VSIm now comes with more than 30 data analysis executables. : See Analysis Scripts
- Standardization of all analyzers so that user defined input will show up as entry boxes.

5.5.5 VSIm Graphical User Interface (VSImComposer)

- Easy, configurable tree-based/visual setup for many problems : See Visual-setup Simulations
- Ability to import materials from a file. : See Importing Materials
- Easily switch between different simulation types (electrostatic, electromagnetic, with or without particles) with GUI adapting to the selection. : See Basic Settings
- Import CAD objects from multiple formats (step, stl, ply, vtk) and set materials of those objects : See Geometries
- Multiple Geometry shapes can be visually added to simulation. : See Geometries
- Ability to define parts through constructive solid geometry. : See Geometries
- Ability to define expressions and parameters and use those in defining the simulation. : See Parameters and See Functions and : See SpaceTimeFunctions
- Ability to assign materials to parts, regardless of how they were imported or constructed. : See Importing Materials
- Minimize data reloading during visualization.
- Uniformization of output across all platforms.
- Multiple simulations can be saved to one directory.

5.5.6 Other

- Over 373 use issues fixed.

5.6 Known Issues in VSIm 8.0

- Remote VSImComposer does not work
- EM Cylindrical not implemented for visual setup
- Occasionally, for parallel runs, the stl reader can spuriously set some regions outside of objects to being inside. When this happens, it is manifest in the visualization of the corresponding geometry field in Data Overview. In all cases observed so far, this has been fixed by offsetting the grid a small amount, changing the number of cells, and/or changing the number of parallel processes.

5.7 Deprecated Features

5.7.1 VSim Computational Engine (Vorpal)

The following engine features are deprecated as of the VSim 8.0.0 release and may no longer be supported in future releases of VSim:

Attributes Deprecated in Version 8.0

- The kinds `bitRevDensSrc`, `bitRevDensSrcVW`, `gaussDensSrc`, `gridDenSrcVW`, and `planarPtclEmitter` of the **ParticleSource** block have been deprecated.
 - Please use the `xvLoaderEmitter` kind instead.
- `kind` is deprecated for the **Decomp** block. The regular decomposition is the only one available and therefore this attribute is not needed at all.
 - Please simply omit “`kind =`” from the **Decomp** block
- `stCadRgn` kind of **STRgn** block has been deprecated.
 - Please use `gridRgnBndry` kind of **GridBoundary** block instead. The `gridRgnBndry` is a much faster method for importing STL geometries.
- `cell` kind of **Species** block has been deprecated.
 - Please use any of the remaining kinds listed under `species-kinds`

Features Deprecated in Version 8.0

- Chinese is no longer supported for **VSimComposer**

Attributes Deprecated in Version 7.0

- The attributes `GridDenSrc`, `CoordProdGridPosGen`, `CoordProdGridDenSrcVW` of the **ParticleSource** block have been deprecated.
 - Please use `gridPosGen` of the `xvLoaderEmitter` kind instead.
- The attribute `numPhysCells` of the **Grid** block has been deprecated.
 - Please use `numCells` instead.

Attributes Deprecated in Version 6.0

- The attributes `kind=yeeEmField` and `kind=exp24EmField` of the **EmField** block have been deprecated.
 - Use **MultiField** block instead
- The attributes `kind=block`, `kind=cosSqPulse`, `kind=expPeak`, `kind=gaussianGrad`, `kind=gaussianLapl`, `kind=linearRamp`, `kind=mask`, `kind=muWaveMode`, `kind=periodicSTFunc`, `kind=qFormPolyn`, `kind=radSymFunc`, `kind=sawtoothWave`, `kind=strap` and `kind=waveguide` of the **STFunc** block have been deprecated.
 - Use `kind=expression` and specify desired functional expression instead

- The attributes `kind=smoothWide1D` and `kind=spaceScalarFieldFuncUpdater` of the **FieldUpdater** block have been deprecated.
 - Use `kind=smooth1D` instead
- The **Component** sub block of the **EmField** of `kind=funcEmField` has been deprecated.
 - The components are now set using **STFunc** blocks with the names E0,E1,E2,B1,B2 and B3.

Attributes Deprecated in Version 5.2

- The **STFunc** block with the name **loadProb**, which is a sub-block of the **ParticleSource** block, which is, in turn, a sub-block of the **Species** block, has been renamed to an **STFunc** block with the name `macroDensFunc`.
- The attribute `loadInGuardCellFlag` of the **ParticleSource** block with `kind=ptclEmitter`, which is a sub-block of the **Species** block, has been renamed `loadInGuardCell`.
- The attributes `kind=speciesTrackTagInternals`, `kind=speciesTrackTagTraj`, `kind=speciesTrackTraj`, and `kind=speciesTracVel` have been deprecated
 - Use `kind=speciesTrackTag` instead. We are now only supporting history tracking if the user uses tagged particles.

Attributes Deprecated in Version 5.0

- The components attribute in the **FieldUpdater** blocks of `kind=gpuSTFuncUpdater`, `kind=STFuncUpdater`, and `kind=unaryFieldOpUpdater` has been renamed `writeComponents`.
- The attributes `kind=nonRelESCell` and `kind=2ndOrderRelBorisCell` of the **Species** block have been deprecated. The same models can be accessed with `kind=cell` and `pusher=nonRelEs` for `kind=nonRelESCell` and `kind=cell`, `pusher=relBoris` and `stencil=spline2ndOrder` for `kind=2ndOrderRelBorisCell`.

Blocks Deprecated in Version 5.0

- **Collision**
- **ImpactCollider**
- **Ionizer**
 - Use the Monte-Carlo framework (i.e., the **MonteCarloInteractions** blocks) in place of using the above deprecated blocks.

Attributes Deprecated in Version 4.2

- The attribute `function` of the **BoundaryCondition** block and the **EmField** blocks of `kind=yeeStaticElecField`, `kind=funcEmField`, and `kind=yeeStaticElecFieldTrillinos`, has been replaced with an **STFunc** block with the name **function**.
- The **UserFunc** block now requires specific names. Arbitrary names for this block have been deprecated.

Attributes Deprecated in Version 4.0

- The attribute `periodicityDirs` of the **Decomp** block has been renamed `periodicDirs`.
- The attribute `indices` of the **BoundaryCondition** and **InitialCondition** blocks has been replaced with the attribute `components`.
- The attribute `function` of the **STFunc** block has been replaced by the attribute `kind`.
- The attribute `singleEmission` of the **ParticleSource** block, which is a sub-block of the **Species** block, has been deprecated. The option `singleEmission=true` has been replaced by the option `ptclCountType=noCounting`, while the default setting is equivalent to the option `singleEmission=false`.

Attribute Deprecated in Version 2.0

- The attribute `xtendUpdate` of the **FieldUpdater** block has been replaced by the attributes `cellsToUpdateAboveDomain` and `cellsToUpdateBelowDomain`.

5.7.2 VorpableView

- VorpableView, an IDL-based visualization tool, is a deprecated feature as of VSim 6.0.0.

-
- VorpableView™ © 1999-2002 University of Colorado. All rights reserved.
 - VorpableView™ © 2002-2017 University of Colorado and Tech-X Corporation. All rights reserved.
 - VSim™ except for VorpableView™ is © 2012-2017 Tech-X Corporation. All rights reserved.

For VSim™ licensing details please email sales@txcorp.com. All trademarks are the property of their respective owners. Redistribution of any VSim™ input files from the VSim™ installation or the VSim™ document set, including *VSim Quick Start*, *VSim In Depth*, and *VSim Reference*, is allowed provided that this Copyright statement is also included with the redistribution.