What’s new ut2k10.1.10 May 1, 2016

1. **Fixed issue with vdump compress not working correctly. The vdump compression routine was changed to use a different compression routine to allow a correct compress operation.**

What’s new ut2k10.1.9 April 3, 2016

1. **Fixed issue with the Analog_Functions window.
   Issue was caused by GCC 4.4.7 with a new rule for “strict aliases” that was incompatible with earlier versions of GCC. Issues was fixed by using a flag added to GCC 4.4.7 to provide for compatibility with prior versions of GCC.**

What’s new in ut2010.1.8.1 March 15, 2016

1. **Added 2 new library files to the directory libs in the software distribution, for Centos 6 systems for “dot” for system that did not have these libraries when booted, libexpat.so.0 and libltdl.so.3.**

2. **Added the Perl function, “AddMarker” to exports in the Undertow.pm file.**

What’s new in ut2010.1.8 July 12, 2015

1. **Fixed core dump when using drop and drag to drag a signal name from waveform window signal area to the source code window. Issue was delay in X software to place software into a position to accept mouse move actions after selection of a signal name. This delay was removed.**

2. **Fixed core dump when loading 3 GB file, issue was nulls in signal data were causing the waveform reader software to core dump, nulls in data will no longer cause waveform data reader to core dump, but nulls are in signal data are illegal.**

3. **Added a utdbg binary to the release software distribution in case user gets core dump they can use the utdbg to make a core file Veritools can use to find issue**

4. **Added a SuSe 11.3 port, both 32 and 64 bit ports to software distribution.**

What’s new in ut2010.1.7 May 12, 2015

1. Finished Centos 6 port for both ut201.1.7_32 bit and ut2k10.1.7-64 bit ports.

This port includes builds with all new libraries, and the rebuild of ut, vdump, API routines, Caliper, and PLI routines.

Here are the libs from ldd ut on Centos6_32

```
%ldd ut

        linux-gate.so.1 =>  (0x00cc4000)
        libwlf.so => /usr/lib/libwlf.so (0x00783000)
        libperl.so => /usr/lib/libperl.so (0x00110000)
        libXp.so.6 => /usr/lib/libXp.so.6 (0x00237000)
        libXm.so.4 => /usr/lib/libXm.so.4 (0x003e9000)
        libXmu.so.6 => /usr/lib/libXmu.so.6 (0x00346000)
        libXt.so.6 => /usr/lib/libXt.so.6 (0x030b3000)
        libXext.so.6 => /usr/lib/libXext.so.6 (0x0023f000)
        libX11.so.6 => /usr/lib/libX11.so.6 (0xb7664000)
        libSM.so.6 => /usr/lib/libSM.so.6 (0x00251000)
        libICE.so.6 => /usr/lib/libICE.so.6 (0x00259000)
        libXpm.so.4 => /usr/lib/libXpm.so.4 (0x00273000)
        libcrypt.so.1 => /lib/libcrypt.so.1 (0x00284000)
        libtcl8.5.so => /usr/lib/libtcl8.5.so (0xb7542000)
        libdl.so.2 => /lib/libdl.so.2 (0x002b4000)
        libnsl.so.1 => /lib/libnsl.so.1 (0x002b9000)
        libpthread.so.0 => /lib/libpthread.so.0 (0x002f0000)
        libstdc++.so.6 => /usr/lib/libstdc++.so.6 (0x03bc3000)
        libm.so.6 => /lib/libm.so.6 (0x0030b000)
        libgcc_s.so.1 => /lib/libgcc_s.so.1 (0x00dc6000)
        libc.so.6 => /lib/libc.so.6 (0x00e29000)
        libXft.so.2 => /usr/lib/libXft.so.2 (0x002d4000)
        libjpeg.so.62 => /usr/lib/libjpeg.so.62 (0x04227000)
        libpng12.so.0 => /usr/lib/libpng12.so.0 (0x0035e000)
        libfontconfig.so.1 => /usr/lib/libfontconfig.so.1 (0x00386000)
        libutil.so.1 => /lib/libutil.so.1 (0x002e9000)
        libXau.so.6 => /usr/lib/libXau.so.6 (0x002ed000)
        libxcb.so.1 => /usr/lib/libxcb.so.1 (0x00695000)
        libuuid.so.1 => /lib/libuuid.so.1 (0x00ce5000)
        libfreebl3.so => /lib/libfreebl3.so (0x006b5000)
        /lib/ld-linux.so.2 (0x003c9000)
        libfreetype.so.6 => /usr/lib/libfreetype.so.6 (0xb74a6000)
        libXrender.so.1 => /usr/lib/libXrender.so.1 (0x00335000)
        libz.so.1 => /lib/libz.so.1 (0x00704000)
        libexpat.so.1 => /lib/libexpat.so.1 (0x00718000)
```

Here are the libs from ldd of ut on Centos6-64

```
%ldd ut

linux-vdso.so.1 =>  (0x00007fff0e7ff000)
```

```
libperl.so => /usr/lib64/libperl.so (0x0000003dfc200000)
libXft.so.2 => /usr/lib64/libXft.so.2 (0x0000003d38600000)
libjpeg.so.62 => /usr/lib64/libjpeg.so.62 (0x0000003d3ce00000)
libpng12.so.0 => /usr/lib64/libpng12.so.0 (0x0000003d2ea00000)
libXmu.so.6 => /usr/lib64/libXmu.so.6 (0x0000003d34600000)
libXt.so.6 => /usr/lib64/libXt.so.6 (0x0000003d3f200000)
libXext.so.6 => /usr/lib64/libXext.so.6 (0x0000003d2de00000)
libX11.so.6 => /usr/lib64/libX11.so.6 (0x0000003d2d600000)
libXpm.so.4 => /usr/lib64/libXpm.so.4 (0x0000003d39200000)
libXp.so.6 => /usr/lib64/libXp.so.6 (0x0000003d2aa00000)
libtcl8.4.so => /usr/lib64/libtcl8.4.so (0x0000003369c00000)
libdl.so.2 => /lib64/libdl.so.2 (0x0000003d2ae00000)
libpthread.so.0 => /lib64/libpthread.so.0 (0x00007f74c8d61000)
libstdc++.so.6 => /usr/lib64/libstdc++.so.6 (0x0000003d31600000)
libm.so.6 => /lib64/libm.so.6 (0x0000003d2b200000)
libgcc_s.so.1 => /lib64/libgcc_s.so.1 (0x0000003d30e00000)
libc.so.6 => /lib64/libc.so.6 (0x0000003d2a600000)
libfontconfig.so.1 => /usr/lib64/libfontconfig.so.1 (0x0000003d2ee00000)
libresolv.so.2 => /lib64/libresolv.so.2 (0x0000003d2c600000)
libnsl.so.1 => /lib64/libnsl.so.1 (0x0000003d3ae00000)
libcrypt.so.1 => /lib64/libcrypt.so.1 (0x0000003d3a200000)
libutil.so.1 => /lib64/libutil.so.1 (0x0000003d3be00000)
libfreetype.so.6 => /usr/lib64/libfreetype.so.6 (0x0000003d2e200000)
libXrender.so.1 => /usr/lib64/libXrender.so.1 (0x0000003d2f200000)
libz.so.1 => /lib64/libz.so.1 (0x0000003d2ba00000)
libSM.so.6 => /usr/lib64/libSM.so.6 (0x0000003d32200000)
libICE.so.6 => /usr/lib64/libICE.so.6 (0x0000003d32a00000)
libxcb.so.1 => /usr/lib64/libxcb.so.1 (0x0000003d2d200000)
libXau.so.6 => /usr/lib64/libXau.so.6 (0x0000003d2da00000)
/lib64/ld-linux-x86-64.so.2 (0x0000003d2a200000)
libexpat.so.1 => /lib64/libexpat.so.1 (0x0000003d2e600000)
libfreebl3.so => /lib64/libfreebl3.so (0x0000003d39e00000)
libuuid.so.1 => /lib64/libuuid.so.1 (0x0000003d31a00000)
```

If any of these libs are missing, use yum to get them from the internet.

For example if you are missing libXt.so.6, do the following:

```
%yum install libXt
```

If that did not successfully install the libXt library, then do:

```
%yum reinstall libXt
```

All of the libraries for Centos6-32 and Centos6-64 have been added into the software distribution directory into the directory, “libs”. Users can just add this location to their .cshrc directory as follows:

```
%setenv LD_LIBRARY_PATH $UT_ROOT_DIR/libs
```

2. Fixed issue with the ut starting up in a directory that is not writable. Ut will now check the directory where the ut is started to make sure it is writable and if it is not, issue a notice that the ut start directory must be made writable or that the user must set an environment variable, UT_WRITE_DIR to a writable directory so ut can write out files it needs for correct operation.

3. Examined issue with the utF files that have wrong data for some signals from the FineSim analog simulator. The issue is that these utf files have multiple time points for the same signal that are at the same time. Users linking into the Veritools API must make sure no time points for the same signal are duplicated. Examples of good and incorrect data files for analog utF data are detailed in the software distribution, in API/read_write_utf/analog_example. This file has a README on what the data and time points must look like, and how to turn utf fast files into text files so users can see the data in any utf file.

What’s new in ut2010.1.6

1. Fixed issues in reading HSpice files, hspice.ac0 and hspice.tr0 files. Issues were file boundary conditions that were unknown before.

2. Coming, Centos6 port.

What’s new in ut2010.1.5

1. Speeded up the API for analog simulators by 7-10 times, while reducing the memory foot print by over 60 times for cases with large numbers of sweeps. In general speeds up API by 2 X.

Veritools will place on the Veritools web site, www.veritools.com, the version number for the simulators that will have this new API

What’s new in ut2010.1.4

1. Fixed error in utF files with the logic names

2. Added in Perl5.8.8 so users can use a later version of Perl. Users on 64 bit systems may need to add in the following environment variable:

%setenv LD_LIBRARY_PATH /usr/lib64/perl5/5.8.8/x86_64-linux-thread-multi/CORE This location is the location of where your native Perl is located.

3. Added environment time out so users can set their own time out in vWave now to whatever time they want before vWave gives back the flexlm license.

%setenv VWAVE_TIME_OUT time_in_hours

4. Added an HSpice measure program Caliper stand alone to run measure scripts in batch mode. Users can now run their measure scripts on HSpice results an unlimited number of times in either interactive and batch mode. Note: user needs a license with the feature "caliper". There is currently no charge for this license. For running "Caliper" see the "caliper_examples" directory in the "examples" directory in the software distribution.

5. Added a GUI caliper window Tools -> Caliper, so user can run Caliper with this window within vWave (Undertow). Users can use either the internal caliper library or the external caliper binary from inside vWave by setting the following environment variable:

%setenv UT_CALIPER_EXTERNAL 1

If not set the internal caliper library will be used when running Calipers scripts.

NOTE: if some input files for caliper (.meas file, dir file etc), are not found in PWD, these files will then be searched in UT_WORK_DIR and UT_DUMP_DIR.

What's new in ut2010.1.3+1

1. Fixed a minor issue with the State Diagram window caused by adding in the ULTRA mode for analog features.

2. The program that had been called vdump2 in prior releases is now called vdump. The program vdump2 with a "-Z" option to break the resultant fast files into many virtual files, had been created to get around a speed issue with Redhat5/CentOS5 when creating fast files from very large files. Using older version of Redhat5/CentOS5 this was very slow, a 4 minute conversion on Readhat3/CentOS3 could take hours on the Redhat5/CentOS5 systems. However the combination of a fix added to the vdump and with the very latest version of Redhat5/CentOS5 that has a fix for this disc speed issue, has fixed this issue.

The version of Redaht5/CentOS5 that has this disc speed fix are CentOS5-64 2.6.18-92.el5xen and above.

You can get your version number by using the following command:

```
%uname  –a
```

If you want to upgrade to this latest version do the following:

In super user mode type for CentOS5;

```
%yum update
```

For Redhat5 type;

```
%up2date
```

Then using these later versions of the Redhat5/CentOS5 operating systems, you will use the command;

%vdump –Cxn hsim.out to convert this hsim.out or any file that needs to be converted to our fast format.

A 1.5 GB file to test this conversion speed, gets converted to a 240MB file in 2:28 MIN:SEC using the 64 bit vdump on a 64 bit CentOS5 system and gets converted to the same size file in 2:57 MIN:SEC using a 32 bit vdump on the same 64 bit system. This conversion took 4:44 MIN:SEC on a 32 bit Redhat 3 system. Hence, conversions should be almost 2 X faster now on the 64 CentOS-5 systems with the latest version of the CentOS-5 operating system.

What's new in ut2010.1.3

1. Expanded the ULTA mode to now include both analog signal data from analog simulators and digital data from digital simulators. Users can now speed up loading and displaying analog data by 3-4 times while at the same time reducing their memory foot print when loading and displaying signal by 3-4 times.
Users with mixed mode designs, that have both analog and digital data can now also speed the loading and displaying of signal data by 3-30 times while reducing their memory foot print at the same time by 3-30 times.

2. Added WGL and STIL file translators to the software distribution for vWave and VeritoolsDesigner. Users can now bring their WGL and STIL files into the vWave, while viewing their input/outputs in a schematic view and while also displaying the source code for their WGL and STIL files. Users can modify their WGL or STIL files to fix an issue, by selecting an editor right on the source code window, and then instantly see what effect this source code modification has on the resultant test vector waveforms.

Users can go to the visual-dft directory in the software distribution to find a readme.txt on how to set up and use the new WGL and STIL file translators.

What's new in ut2010.1.2

1. Released the ULTRA mode for digital designs. This update also includes a 3-30 times speed up of loading and displaying digital waveforms into vWave or VeritoolsDesigner from the simulation waveform data files. The memory footprint for loading and displaying waveforms is now also 3-30 times smaller.

NOTE: Users who want to use the ULTRA mode to speed up loading and reading data into vWave and VeritoolsDesigner must first obtain a license for ULTRA, listed on the license feature line as, "ut_ultra_feature". These licenses are provided at no cost to users who are currently on maintenance and will remain in effect while the user stays current on maintenance.

2. Added the ULTRA_PLUS mode of operation, which removes unconnected and redundant signals from the output simulator data. This can provide a many times speed up of the simulator while resulting in smaller waveform files, smaller by many times.

NOTE: Users who want to use the ULTRA_PLUS feature to speed their simulator speeds must first obtain a license for ULTRA_PLUS, the license lists this feature as the "ut_ultra_plus_feature".

Users can use both the ULTRA to speed up loading and displaying signal data, while at the same time using the ULTRA_PLUS feature to speed up their simulator.

3. New system commands were added to allow users running the VCS simulator to run three times faster when the user combined Verilog with SystemVerilog in their design. This will speed up any VCS simulation running Verilog and SystemVerilog by 3 times. This feature is provided at no cost to users who are currently on maintenance and will remain in effect while the user is current on maintenance.

4. Fixed issue with API and using double precision numbers so waveforms from these files will now look like analog and signals in these type of files can now be interpolated.

5. Fixed issue with vdump2 where the data sometimes would go out of scale.

What’s new in ut2010.1.1

1. Fixed issue with API and using double precision numbers so waveforms from these files will now look like analog and signals in these type of files can now be interpolated.

What’s new in ut2010.1.0

1. Completed the “ULTRA” feature, this makes the software 4-40 times faster when loading and digital displaying signals by allowing users to have a much more compact data base.

2. Added SUSE 11.2 support for 32 and 64 bit platforms

3. Updated User Guide ut2010.1.0 to indicate that based on testing that the fastest commands for VCS, when using Verilog with SystemVerilog are;

```
$dumpvars():
$vtDump;
```

This combination of commands gives simulation speeds that are about 2 ½ to 3 times faster, than using routines that are using the VPI system calls.

What's new in ut2008.1.4

1. Fixed hierarchically connected vector signal waveforms

2. Fixed hierarchical name listing for any signal

What's new in ut2008.1.3

1. Added major speed up to the Veritools VPI, PLI by removing callbacks for signals that are unconnected or do not change.

2. Added additional speeds ups by optimizing VPI code

3. Added SuSe 32 bit port

4. Added CentOS-5 both 64 bit and 32 bit ports

5. Added a libvtvpi_vcs.so port for Verilog only VCS

6. Fixed static declaration of memory file calls to work with VCS simulator.

7. Veritools work around for single +incdir+ command line option, use multiple +incdir+ command line options if user wants more than one +incdir+:

   ```
   +incdir+dir1 +incdir+dir2 +incdir+dir3
   ```

8. Built the PLI for the Aldec Riviera-PRO simulator

9. Fixed cosmetic spacing issue for vWave/Undertow for utf writers that display the module names too close together in the choose window, shows up when sub-modules have short names that are exactly the same length

10. Added support to utF API so utF files can go above 2147484148 bytes

What's new in ut2008.1.2

1. The schematic now makes room for updating blank values on component pins

2. Fixed dragging single items from the choose window to single line text areas such as the eye diagram window and the convert to digital window.

3. Added "Standard eye with clock signal" to the eye diagram window.

4. Fixed snap for eye diagram waveforms.

5. The new vdump executable is called vdump2. (A new ut executable is required to read the new vdump2 files.) It supports two new command line arguments:

-Z #chunks

This tells the vdump2 to make a 280 mode file using the specified number of chunks equivalent to +VTNUMCHUNKS in the PLI. The default is 1 which tells it not to make a 280 mode file.

-z #MB_per_chunk

This tells the vdump2 the size to use for a compression chunk in megabytes which is equivalent to +VTCHUNKSIZE in the PLI. The default is 16 which tells it to use 16MB per chunk.

6. Added support for making uncompressed fast files over 2GB in size with vdump2.

WHATS NEW IN 2008.1.1

1. Added support for modeltech fli in libvtvpivhpi_mt.so.

2. "Display Current Drivers" in the source window now uses the Expression evaluator to try to find the current driver for the signal.

3. Added several partial update strategies to the schematic window to keep it from slowing down the whole tool. Prior to this feature moving the cursor on the waveform window would be slow if a large schematic with many elements was displayed due to the speed of updating all of these elements.

4. Verified support for SV interfaces and SV modports.

5. Fixed highlighted nets on the schematic so they are retained when going up scope and down scope.

6. Decoupled module port names from the pins outside of the module so they can be used for selecting restricted nets on down scope in the schematic without affecting other features (highlight net, etc.)

7. Up scope in the schematic will now automatically expand nets that were previously highlighted.

8. Added Options -> Up scope, Show Whole Module to the schematic window. When toggled "on", the Up scope operation will show the whole schematic at the new scope.

NOTE: Up scope operations: Users can now select a net on a pin of any element going out of this module, and go up scope to see this module and pin and the connections to other gates and modules connected to this pin.

Toggling down the Option => Up scope, Show whole module will show the entire schematic for the up scope module and the selected net in the lower module will be highlighted in the up scope module.

Down Scope Operations: Users go down scope by double clicking into the black space inside of any module. If an outside pin on this module is selected, the down scope module will highlight the pin that was selected in the up scope module. If a module inside pin is selected, then the user will see only the down scope schematic for that selected pin and the logic connected to just that pin.

WHATS NEW IN 2008.1.0

1. Updated the VPI/VHPI to include MTI FLI interface.

2. Added latest parser to vtcom and VeritoolsDesigner

3. Added in trace back for Show Input Cone for schematics with no signal values.

4. Added in trace back for tracing from point A on schematic to point B, an instance name of schematic element.

5. **New PLI commands, modes:**

$vtVpiDumpflush;

$vtDumpSuppress(name), The "name" can either be a hierarchical module name or a defined module name.

$vtVpiDumpSuppress(name), The "name" can either be a hierarchical module name or a defined module name.

$vtDumpsuppress/$vtVpiDumpsuppress, calls must be made before the call to
$vtDumpvars/$vtVpiDumpvars in order to have any effect.

Using the new +VTCOMPRESS280 mode to make compressed files.

This file format divides the data into a fixed number of separate chunks. +VTNUMCHUNKS<number> controls the number of chunks to use.

Only the first $vtDumpvars() call will be divided into separate chunks. Subsequent $vtDumpvars() calls will each use one new data chunk. +VTNUMCHUNKS10 is the default. +VTCHUNKSIZE<number> can be used to set the size of each uncompressed data chunk in MB. +VTCHUNKSIZE16 is the default (16MB). Examples:

```
%ncsim -f ncsim.args +VTCOMPRESS280

%ncsim -f ncsim.args +VTCOMPRESS280 +VTNUMCHUNKS200

%ncsim -f ncsim.args +VTCOMPRESS280 +VTNUMCHUNKS1000
+VTCHUNKSIZE4
```