1 Release Notes 2.9.5

1.1 Library

- Added new Resource_FCFS_N_Shared block
- Added Write capability to the Database block
- Made the statistics reporting for Schedulers, Time_based, Queue and Channels. The collect and set start at 1 and not at 0.
- Added capabilities for binary string data type to be processed in all DS_* blocks
- Added a queue on the input port of the DS_While_K block
- Added new block called stats_sum to accumulate data at a certain point in the model
- Added relative path to the File Selection parameter fields of all blocks.
- Added additional error statements

1.2 Documentation

- Updated documentation for all blocks with use_models and refined explanation
- Added the new demonstrations into Example list

1.3 Demonstration Systems

- Unmanned Aerial vehicle
- RP3
- Hardware-Software partitioning for a complex electronic system with a variety of use cases
- Designing off-chip Frame Buffers
- Using C interface for sending data to a VisualSim output port

2 Release Notes 2.9.1

2.1 License Manager

- Tested on Windows, Linux and Solaris platforms
- Reduced memory consumption by about 40%
- Added library licenseing
- Added additional statistics reporting to the log file

2.2 Library

- Library structure has been completely reorganized. Existing users, please refer to Chapter 5 of the documentation for the mapping between the original location and the current location of blocks.
- Added new blocks to work with XML- Traffic, Database and DS_to_XML
- Added new blocks to create traffic from a source or trace file- Source->Traffic->Traffic

- Added new block to create a lookup table with generic list of columns and flexible compare utility
- Modified all library blocks to accept both strings and Record Token
- Eliminated the need for Tieoff of unconnected input ports
- Data Types for all block ports are now polymorphic
- Added a flow-through mechanism for both Channel_Complete and QS_Free
- Added simplified mapping block- Task_Basic
- First availability of block-based Architecture library
- Increased data type support and additional compilers for the C++ Application interface
- Visual identification of the virtual named mapping between the Task_Issue/Task_Basic and Schedulers and between Virtual Conenction blocks
- Improved icons for all blocks that are descriptive of the internal operations

2.3 Documentation

- Chapter 1 has been modified to explain the modeling technique and matched to the new library structure
- Installation document has detailed description of the licensing options
- Chapter 5 has now the new organization of the block documentation
- The on-line block help has been improved for easier reading

2.4 Example Systems

- Added new example for Automotive- FlexRay bus, and Rollover Prevention
- Added example for hardware design

2.5 Training Material

- Added 1 hardware and 1 software mapping example to the tutorial
- Modified training material for the new library structure

3 Release Notes 2.8.0

3.1 License Manager

- Floating and server license is now available
- Mobile License manager added

3.2 Library

- Sim_Monitor: This block updates the time of the simulation at equal intervals
- STK_Intf: This makes a single connection to STK Toolkit for a co-simulation.
- STK_Port: This block accepts the commands that need to be transmitted to STK and generates the response string on the output.
- uEngine: This block is used to construct the script for the SmartMachine

3.3 Documentation

Documentation has been extended in the introduction, description of the SmartMachine and the interfaces

- Chapter 1 now contains a section on selecting the right simulator
- Chapter 6 now contains the Java, C++ and Applet information only
- Chapter 8 now contains the interface information for Tcl, MatLab and STK
- The Index Page has explanations on the different models of computation
- Added description on the 3G/ATM router demo, Moving mobile handset design and Functional cache

3.4 Example Systems

- ATM Router Processor Blade server- Mapping software tasks on architecture
- Functional cache- Determines the optimal cache configuration for an MPEG Decoder executing on an ARM 13 pocessor. The ARM processor is not available with this model and a trace file is shipped.
- Cellular Phone Platform- Analyze the task processing on the cellular platform and optimize operations for power and utilization
- Mobile Cellular phone- Software estimation on multiple hardware architectures
- TCP header compression model

3.5 Training Material

- Added information on software architectures, Modal Model, SmartMachine
- Updated new methodology for bus, processor and memory modeling

4 Release Notes v2.5.5

4.1 Library

- This is the first release of the VisualSim SmartMachine.
- In addition some updates from intermediate release have been compiled together.
- Statistics in the Resource_FCFS have been upgraded to add Utilization.

4.2 Documentation and Front Page

The Front Page and Documentation has been reorganized one more time.

- The Step by step excercises for the block model and FSM are under Training.
- The documentation has been consolidated and the flow has been improved. A methodology section has also been added.
- The layout of the documentation has been modified. Please the overview document to see the changes
- The style has been modified to follow a technology-centric rather than tool-centric flow.

4.3 Example Systems

- VME Bus System
- ABS Braking System
- Parallel Processor
- Switch model using SmartMachine
- 15 new examples for the SmartMachine
- Article showing the modeling the Switch Router model using basic Queues, Resource_N and the new SmartMachine/uEngine blocks.

4.4 Library Reorganizations

- Excution control has been moved to Utilities under Model Library.
- Time_select has been to the Utilities folder in the Model Library
- All blocks under folders marked “Others” are scheduled for deprecation in a future release.
- All the Statistics blocks have been moved into the ‘Statistics’ folder within ‘Display and Statistics’ library. These are Stats_Basic, Stats_Probability, Stats_Difference and Stats_Memory
- All the Text display blocks have been moved under ‘Textual_Display’ folder of the ‘Display and Statistics’. These are Display_Text (Disp_Fast), Memory_Monitor, InteractiveShell, Display_Text (Non-Buffered) and MonitorValue.
- All the existing Viewers and Plotters have been moved to Viewers-Plotters. These are XYPlotter, BarGraph, TimedPlotter, HistogramPlotter, SketchedSource, RealTimePlotter, Recorder, ArrayPlotter, SequencePlotter, SequencedScope, TimedScope, XYScope, MatrixViewer and MonitorImage.
- ‘Time and Delay’ inside ‘Others’ folder has been moved under ‘Block’ Library and renamed ‘Clock Source Delay’. SketchedSource has been moved to Viewers-Plotters folder above.
- Server inside ‘Time and Delay’ has been deleted.

4.5 C/C++ Interface

The C/C++ wrapper function has been greatly extended. Now all data types are supported including char, double, long, short, Record Token, strings, Boolean and one-dimensional arrays of these data types. Asynchronous calls can be made directly from the C code to the ports.

5 Release Notes v2.5.1

5.1.1 Description of the Release Numbers

The first number in the release indicates the version of the product, the second number indicates the library release and the third number indicates bug fix updates.

5.1.2 Library

Over 60 blocks in the library were modified for updates to error messages, syntax checking, performance improvements and statistics.

5.1.2.1 New Blocks

DS_Expression Blocks
There are 5 new blocks in this category and they are located under Block Library->Data Structures->Expressions. These blocks are DS_Expression, DS_Expression_Gen, DS_Expression_Trig and DS_Expression_Delay. These blocks are an advancement to the DS_* blocks and use the extensive VisualSim Expression Language.

Out_Fld_Mem Block
This block extends the basic OUT block to support Fields and Memories to retrieve destination information.

5.1.2.2 Existing Blocks

Resource_QS Blocks
The QS blocks now support the incoming DS to be output on the DS_free_output port. These blocks also now have non-blocking and blocking mode for the incoming requests. The Lowest_Priority reject mechanism in the Allocate block has been removed.

Resource_N Blocks
The Resource_N will also support Scan operation when the quantity in the index is zero. An error message will be returned if an operation is attempted on a field where no DS exists in the Index.

Scheduler Blocks
The Scheduler_HW expansion port will now output the DS arriving from the Task_Issue block. The Scheduler_SW will place the DS from the Task_Issue block on the extension port at the conclusion of every processing.

DS_Switch_K/ DS_Select_K/ DS_Select_Port Block
These will now output the values in a specified Memory location on the switch/select_port.

All Blocks

- There are now a total of 36 type conversions added to the DS_Expression_Language.
- Checks to make sure there are no name overlap between parameters, fields and memories

- Checks to eliminate overlap in destination names between Schedulers and Virtual Connections.

5.1.2.3 Blocks Reorganization

- The IN, OUT, MUX and DEMUX blocks have their own section and are now under Library: Digital->Virtual Connections.
- The Memory_Monitor block has been to the moved Display and Statistics Folder from the Memory Folder.
- DLY, Const and Tieoff have been moved to Model Folder->Utility. For the current release it is also available in the Virtual Connection and Switch Folder
- DS_Build_Composite, DS_Add_Sub_Field, DS_Insert_Array and DS_Select_Array have been moved to Block->Data Structures->Unsupported. This is has been scheduled for removal with 3.0.

5.1.3 Demonstrations

- MAC Layer for the 802.11
- MAC Layer of the Bluetooth Protocol
- Resource Modeling over 802.11 system
- SPI3 Bus Model
- Aerial Combat Sensor Modeling of aircraft and other large system design
- 1553B multi-drop bus
- 20 additional examples to go with the new documentation

5.1.4 Documentation:

- **Reorganization**: The documentation has been reorganized and enhanced with a number of sections. The flow has been modified to reflect user learning pattern.
- **Search Tool**: A search index has been added to search entire documentation. This is available from Help->Search Help menu of the BDE.
- **Error Message**: Over 350 errors have been updated for more user-friendly message. The new messages include full hierarchy of the block, the specific field and recommendations.
- **Document Access**: The documentation can now be accessed from the Front_Page on the UNIX machines irrespective of the installed Web Browser.

5.1.5 Known Errors

- Icons on the first window opened sometimes smears. This is a repaint problem. The workaround is to open any parameter window and select "Cancel". This does not happen on all machines
- Copy and Paste has a problem where the pasted items are placed on top of the original icon. Also the pasted items are not selected and each pasted icon needs to be moved separately.
- Image Display does not exit when the model has been closed.
- If there are multiple windows open for a model, say, the top level ModelBuilder window and the tree view, or the top level window and the inside look of a composite component, when the top level window is closed, all other windows are closed, but the user is not prompted to save any change to the model

6 Release Notes v2.0.4

6.1.1 Introduction

This release simplifies the block documentation and also adds enhanced capabilities for accessing instances of a dynamic instantiation. This note is incremental to v2.0.3.

6.1.2 Documentation

The block documentation has been simplified to include only the block, parameter and port descriptions. All the summaries and extraneous information that is not required for modeling have been eliminated. The utilities that are available for custom-coded blocks continue to have the complete documentation of the methods and constructors. Documentation of some of the resource blocks have also been augmented with more information. A detailed example system is now available for the Dynamic Instantiation.

6.1.3 Libraries

- Conversions blocks have been added to communicate to and from a specific Dynamic Instance.
- The statistics for all the dimenions or queues of the QS_resource, FCFS_Resource, Queues_N and Virtual_resources can be triggered on the stats_in port by providing a integer value equal to the Max_Queue_Length.

6.1.4 Example Systems

- Resiliant Packet Ring
- SPI-3

6.1.5 Errata

- The Disp_Fast has been renamed to Display_Text (Buffered) in the Library Structure. It is located in the Statistics and Display folder.

7 Release Notes v2.0.3

7.1.1 Introduction

This release has focused on new libraries to simplify traffic creation from files, eliminated compiling of Data Structures and better matching of instantiated XML components. This note is incremental to v2.0.1.

7.1.2 Libraries

- A new block has been added to the SmartBlock->List folder. This block combines Add DS and Lookup into one command. The traffic can be defined either using text field, text file or CSV format in this block.
- FCFS blocks have been updated to allow for concurrent and synchronous events to occure from multiple queues in the same Resource.
- Reduced the internal token generation to speed up simulation.

Modified Library structure

- Less frequently used blocks in the Data Structure library have been moved to the "Others" folder within the Data Structure directory.
- Memory Operators have been renamed to Memory Access to reflect the fact that these blocks can be used to read and write data from a specific location.

7.1.3 Data Structure

- Data Structures can be defined in text file. So thyere is no need to recompile for every change.

7.1.4 XML

- A XML model file can be instantiated directly in the Hierarchical block. Current support is for this feature to be done manually by entering the reference in the XML file of the model.

8 Release Notes v2.0.1

8.1.1 Introduction

This release includes a combination of performance tuning, feature enhancements and block additions/changes. This note is incremental to the Release Notes from v2.0.

8.1.2 Libraries

There have been substantial addition to the libraries. In addition, there has been modification to existing blocks for additional functionality. Please review these before continuing model development.

8.1.2.1 New Library Blocks

- Added binary support in all the DS blocks. This includes Verilog-like nomenclature, bit-wise operators and binary strings.
- Created a new suite of blocks called Resource_N. These blocks access large address space for storing, operating and retrieving data. There are 5 blocks in this new sert and are all located in the Block Library-> SmartBlocks->Resources.
    - Resource_N
    - Resource_N_Write
    - Resource_N_Read
    - Resource_N_Stats
    - Resource_N_Scan
- Added a unbuffered Text Display Viewer- "Display(Non-Buffered)"
- Added the ability to use the parameters inside the "" in the DS_* blocks and in the Resource_N blocks. For all other blocks, the parameters must be outside the "".

8.1.3 Demonstration Systems

1. Router model using the Resource_N blocks
2. Models to describe the use of the binary structures and the Resource_N blocks in the training directory.

8.1.4 Documentation

- Additional documentation for the DS_* blocks
- New blocks have documentation associated
- Updated the Class Material with new information and handling common problems
- Improved format of the block library

8.1.5 Simulator

- Improved simulation performance by about 30% in all the DS_* blocks.

8.1.6 Tool Enhancements

8.1.6.1 General

- Added user documentation fields for Blocks, hierarchical blocks, ports and wires/relations.

9 Release Notes- v2.0.1

9.1.1 Introduction

This release includes a combination of feature enhancements and block additions/changes. This also combines all of the patch updates between version 1.0.1 and 1.4.3u4.

9.1.2 Libraries

There have been substantial addition to the libraries. In addition, there has been modification to existing blocks for additional functionality. Please review these before continuing model development.

9.1.2.1 Reorganization

- The INIT, Const and DLY block are now also included in the Digital->Flow Control library for easy access.
- All the base blocks (Hierarchical, Dynamic Instantiation, C interface, MatLab, FSM and Modal Model) are provided under the base block folder.
- DEMUX and MUX block names have been switched.
- All the simulators are now under the simulator folder in the Model Library.
- Mathematics and, Display and Statistics have been moved to the Block Library
- Traffic Library has been renamed Data Structures

9.1.2.2 New Library Blocks

- Added "Listen to Block" detail for all advanced blocks.
- Key blocks now pass Block Name to common utility that reads parameter names, so user can identify block more easily that may have a parameter issue.
- Base Blocks
  - Dynamic Instantiation is in the Model Library->Base Blocks. This actor creates a parameterized number of instances of a hierarchical block.
- IO
  - The FileWriter and DoubleReader actors have been removed, use the LineWriter or ExpressionWriter actors instead of FileWriter and use the LineReader and ExpressionReader actors instead of DoubleReader.
  - Interactive Shell- This is located in Model Library->IO->io and is used to accept text input from the user when triggered.
- Digital
  - DEMUX- Name has been from MUX
  - MUX- Name has been from MUX
  - EIOx is in the Blocks Library->Digital Library->Flow Control and is a variation of the EIO that does not contain the delay between executions of each output port.
  - Memory_Monitor is in Block Library->Digital->Memory- that allows user to trace a model memory on read or write, based on start, stop time entered by user. Block Name, simuation time provided. There is also an input to read all memory locations, on a separate output port.
  - Connect_EIO is in the Blocks Library->Digital Library->Flow Control and can be used to control the simulation execution.
  - Connect_OR is in the Blocks Library->Digital Library->Flow Control and can be used to control the simulation execution.
  - DS_State_Switch is located in the Blocks Library->Digital Library->Data Structure and is a combination of Case Switch and DS Insert. The selected Field is compared for the matching values in the Value parameter. Based on the match,

  - the respective Field line is executed and then resulting DS is sent out on the respective output port.
  - DS_If_Else_Insert_K is in the Blocks Library->Digital Library->Data Structure and is a variation of the DS_If_Else_K. This has a single output that sends the DS output on irrespective of whether the expression evaluated to a True or False.
- SmartBlocks
  - Model_List_DS: Added to the Blocks->SmartBlocks->Lists. The Model_List can be used only with the ListItem_Class DS. This new List can be used with any DS.
- Mathematics
  - Array Maximum is in the Blocks->Mathematics->Array.
  - Array Minimum is in the Blocks->Mathematics->Array.
  - Rician is in the Blocks Library->Mathematics->Number Generator and outputs a random sequence with a Rician distribution.
- Signal Processing
  - Communication Library (Block Library->Signal Processing) has been added with 6 new functions including Convolution Encoder and Viterbi Decoder.
- Imaging
  - Imaging Library including Java Advanced Imaging and Java Media Framework in the Block Library.\
- Display and Statistics
  - Sketched Source is located in is in the Blocks Library->Displays and Statistics and is a plotter output where the plot lines can be varied to provide a variable input.
  - Image Monitor in the Blocks Library->Display and Statistics.
- Others
  - Execution Control library under the Block Library->Others Folder. This controls the simulation using a 'STOP', 'ThrowException' and 'ThrowModelError'.
  - Prototype library is in the Blocks Library->Others and includes a set of blocks + simulator to define the constructs of a 3D animation. This is based on the Java 3D library and can be expanded using the available methods.

9.1.2.3 Removed

- Memory_basic was an experimental block and has been superseded by the Memory_Token block.
- The FileWriter and DoubleReader actors have been removed, use the LineWriter or ExpressionWriter actors instead of FileWriter and use the LineReader and ExpressionReader actors instead of DoubleReader.

9.1.2.4 Modified

<u>Digital</u>

- DEMUX- The block that used to be called MUX is now called DEMUX and the block that used to be called DEMUX is now called MUX. This is to make it consistent with hardware notations.
- MUX- The block that used to be called DEMUX is now called MUX and the block that used to be called MUX is now called DEMUX. This is to make it consistent with hardware notations.
- Memory_Write- The parameter can accept the name of the memory location or a reference to an incoming field that contains the memory location name.
- EIO now has a small 1/10 of a pico-second delay between the executions of the output ports.

- Queue Blocks added 'pop_queue_input' with ability to select an element other than the front queue element, using an integer, 0 (front), 1 (next), etc.
- DS_Generate now will provide detailed information on missing data structure, readable exception generated.
- DS_Generate, DS_Trigger now supports Data_Structure.txt feature to allow Data Structures to be modified without impacting current simulation, simply save the new changes in *.txt file. DS name also supports a path/file name format.
- DS_* Blocks support string type on DS ports to minimize port type conflicts.
- DS_* Blocks now support the ability to reference composite. Data Structures in memory, using the following syntax:
  MyMemoryName:FieldName (Memory and Field of DS in this memory)
  MyMemoryName::FieldName (Memory references a memory name that contains the DS Memory name and Field of DS in this memory)

  MyMemoryName::FieldName can reference a Data Structure in memory indirectly.
- DS_* Blocks now supports more data type conversions, based on user data types for OP_X and OP_Y/OP_Z:

OP_X		OP_Y/OP_Z (Both same type)
integer	=	double
integer	=	string
long	=	double
long	=	string
double	=	integer
double	=	long
double	=	string
boolean	=	string
string	=	integer
string	=	long
string	=	double
string	=	boolean

- Parse_File, Parse_String not sensitive to order of parse input.

SmartBlocks

- Scheduler output DS has been changed to Task_Class to acommodate additional information. Schedulers can also have a comma-separated list of variables for Task_int_execution_number.
- Task_Issue, Scheduler_SW issues an exception if parent name is non-existant.
- Scheduler_SW and Scheduler_HW now execute context switch time, round robin time based on relative time or number of clocks Time_Type setting.
- Scheduler_SW and Scheduler_HW more accurately count round robin transitions.
- Scheduler_HW can now accept a comma-separated list of internal execution.

Others

- AudioReader now uses a FileAttribute instead of a string valued parameter. The name of this parameter has changed from sourceURL to fileOrURL to conform with LineReader and ExpressionReader.

9.1.3 Demonstration Systems

1. Transmission
2. Imaging Examples
3. Switch and Router
4. Hardware-resource modeling
5. Crossbar chipset architecture

9.1.4 Documentation

- All blocks documentation has been completed updated with details for each port, parameter, block usage and use-case.
- Enhanced training model containing-
  - Brief block overview
  - New training material,
  - Description of commonly used operation, h
  - How-to tasks,
  - New training labs and tutorials,
  - Modeling guide for the digital library and
  - TCL guide

9.1.5 Interfaces

1. C/C++ interface: Interface to C/C++ functions using the Java Native Interface
2. MatLab interface
3. SerialComm link to accept and transmit using the Serial Port of the PC. This is currently supported on Windows only.
4. Keyboard sensor
5. Network-based transfer of information between versions of VisualSim.

9.1.6 Simulators and Data Types

- Continuous Time (CT)
  - Continuous Time (CT) reworked for improved mixed signal and hybrid systems semantics.
  - In the CT (continuous-time), instances of Clock and TriggeredClock very likely should be manually changed to ContinuousClock and TriggeredContinuousClock. The reason for this is that we have made progress on the semantics of signals with discontinuities and have improved the software implementation. We created a new set of clocks for the CT simulator. ContinuousClock and TriggeredContinuousClock generate piecewise constant signals with discontinuities. The discontinuities are correctly handled regardless of step sizes, and threshold crossing detectors properly identify discrete transitions across the threshold. In principle, Clock and TriggeredClock can still be used in CT models, e.g. to generate a sequence of events, but it is better to use the EventSource block for this purpose.
  - In hybrid systems models (CT + FSM), the signal types for modal models may not be inferred as expected. If a modal model has both continuous and discrete signal types at its inputs, it may need manual adjustment of the output signal types. To do this, right click an output port and add a "signalType" parameter; then set the value of the parameter to "CONTINUOUS" or "DISCRETE" depending on your needs.
- FSM and Modal Model

  - Transition Refinements: A transition in an FSM can have a refinement, which is a model that executes when the refinement is taken.
- Synchronous Data Flow
  - We have improved the way that the SDF (synchronous dataflow) domain implements rate propagation. Previously, SDF would infer the rates of ports of an SDF model, and create parameters named "tokenConsumptionRate" "tokenInitProduction" and "tokenConsumptionRate". We've changed this so that it sets these values only if the parameters do not exist already, and stores them in new attributes called "_tokenConsumptionRate" "_tokenInitProduction" and "_tokenConsumptionRate". These are transient (instances of Variable rather than Parameter), so they will not trigger a prompt to save.
- Data Types
  - Function Token: An expression that defines a function can be passed among actors in a function token.

9.1.7 Tool Enhancements

9.1.7.1 General

- Block Icons enhanced to be Web-safe, improved aesthetics and faster screen painting
- Increased simulation performance for DE simulator
- Removed the Memory Monitor
- Run buttons on the toolbar and hotkeys.
- Persistent window sizing and placement.
- Slider Parameter
- An interactive expression shell
- An interactive TCL shell
- Removed the automatic layout function from the Menu and Toolbar.

9.1.7.2 Menu

- Removed XML view from the Menu. This is still available from the Hierarchical View on the Library Tab
- Undo added to the Edit Menu and the Toolbar.
- Redo added to the Edit Menu and the Toolbar.
- Save complete model in library
- Create a hierarchy out of a section of the model
- Removed multiport from the Toolbar. This is available from the Menu bar.
- Reorganized the Help Menu for easy access

9.1.7.3 Context Menu

- Add breakpoints
- Save block in Library
- Use custom icons

9.1.8 Known Errors

- Icon Smearing occurs when VisualSim is first opened. This is a repaint problem. The workaround is to open any parameter window and select "Cancel". This does not occur anytime else. Also, this is not consistent from machine to machine.

- Copy and Paste has a problem where the pasted items are placed on top of the original icon. Also the pasted items are not selected and each pasted icon needs to be moved separately.
- Image Display does not exit when the model has been closed.
- If there are multiple windows open for a model, say, the top level ModelBuilder window and the tree view, or the top level window and the inside look of a composite component, when the top level window is closed, all other windows are closed, but the user is not prompted to save any change to the model.