ALTAIR
ONLY FORWARD

RELEASE NOTES

Altair Activate 2021.2

New Features and Enhancements 2021.2

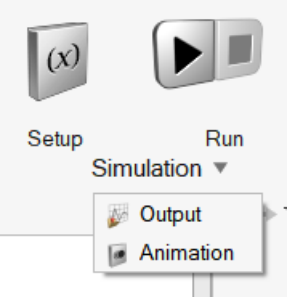
Altair Activate 2021.2 includes the following new features and enhancements.

User Interface

New features include a Simulation ribbon, Extension Manager, and Message Center.

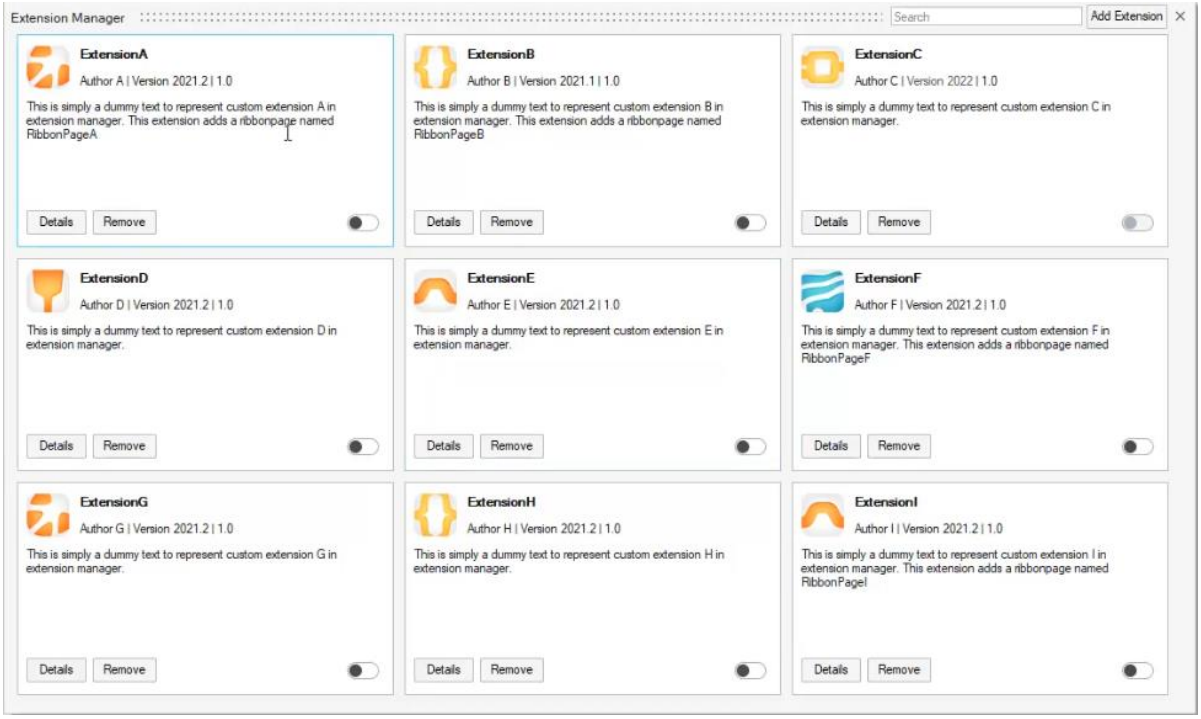
Simulation Ribbon

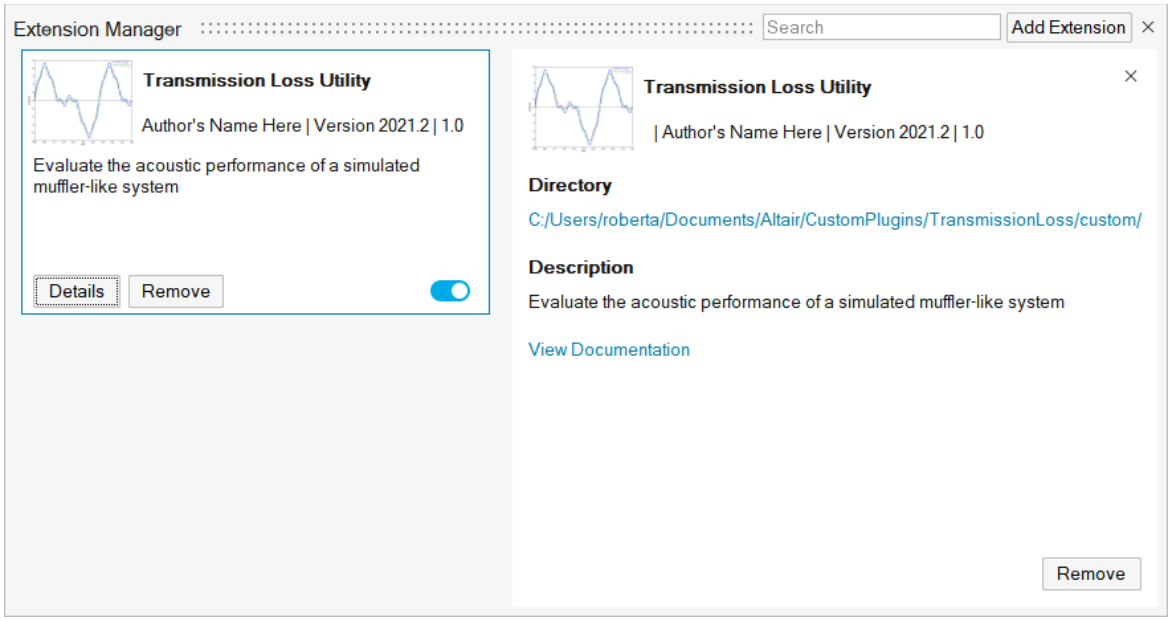
The items in the main ribbon of the user interface have been rearranged for a better user experience. The Simulation ribbon now provides separate buttons for Setup, Output and Animation:



Extension Manager

A new Extension Manager lets you create custom extensions, load extensions automatically, and add custom documentation specific to your defined features.





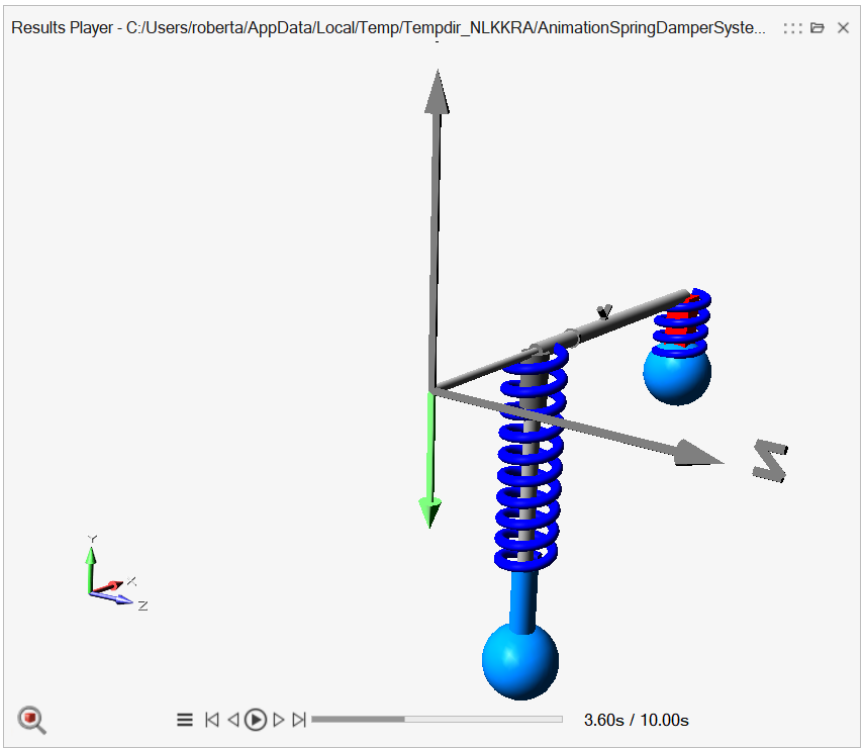
Message Center

Activate error messages and warnings are now displayed and categorized in the new message center window. The existing OML window is now dedicated to OML statements and the evaluation of functions.

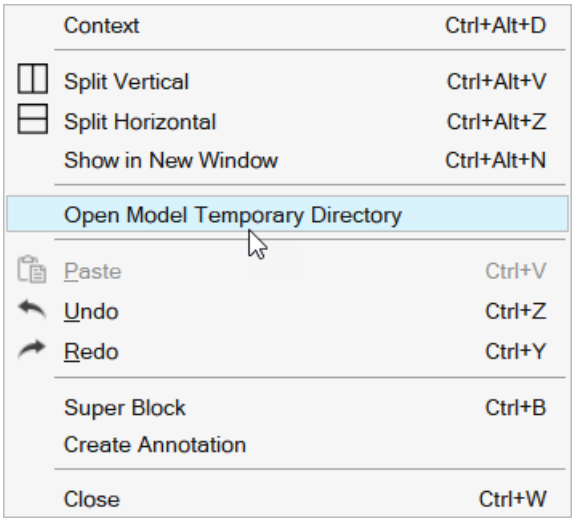
Modelica

Results Player

- The Results Player can now be used to load and animate MTSF files generated by the simulator.



- A new context menu option lets you open the temporary folder and locate where the MTSF file is saved.



- Modelica library import: support of non-structured libraries.
- Improved support for reading MAT files and various performance improvements.

Modelica Compiler Enhancements

- Improved support for reading MAT files and various performance improvements.

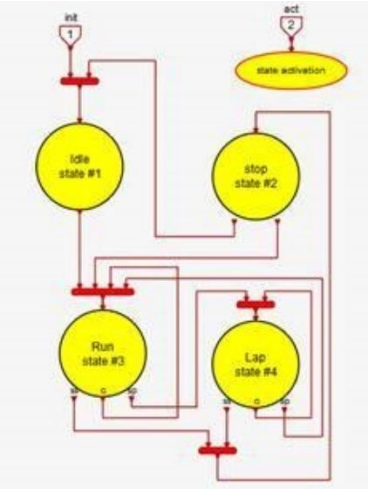
Additional Changes and Enhancements for Activate

Computer Vision Library

Around fifty new OML functions help you derive a vast array of information from digital images.*

New State Machine Library (Beta Release)

This block library provides a methodology for constructing finite state machines. The library supports concurrent state machines and hierarchical constructions. Run-to-completion and history features can be used as well. Fully implemented into Activate, the library lets you leverage all the available features including parameterization and code generation. Documentation and demos are provided.



User Interface

- After dragging a block into the BDE, you can now use the **Del** button to remove the selected block.
- Live printing is now available in the Python command window.

New Blocks & Blocks Improvements

- Expression block parameters: In the expressions of the `MatrixExpression`, `MathExpression`, `MathExpressions` and `IfExpressions` blocks, the auto-masking operation now lets you view the parameters that are not explicitly defined as block parameters. Super Blocks that include these blocks are also properly auto-masked.
- The `MatrixExpression` block now supports integer data types.
- Lookup table blocks: new blocks let you set table parameters as input signals rather than block parameters. This allows the tables to be updated dynamically during simulation.
- The `MotionSolve` block now allows the definition of Motion Solve environment variables.*

- The `FromCSV` block performance has been improved for large files. It also has better support for file separators (including auto detection).
- The `FromMAT` and `FromHDF` blocks now support the Curve Editor.
- New `SmoothStep` block: This block provides a nonlinear smooth step function equal to the Step3 and Step5 MotionSolve functions.
- Scope blocks: an API is now available to get a scope handle. Plot functions can then use this handle.

FMU Export Using TCC

The embedded TCC compiler has been updated and can be used to export FMU's. This compiler is used in Windows when no MS Visual C compiler is found on the system.

Hydraulics by Fluidon Library

The following new blocks are available:

- Components for load sensing applications
- Proportional valves
- PropValve43LS
- PropValve43LSTableAx
- PropValve43LSTableQpx
- Pump controller
- LS controller
- Pressure compensator
- Counterbalance valve (resp. load control valve)
- Pilot controlled check valve

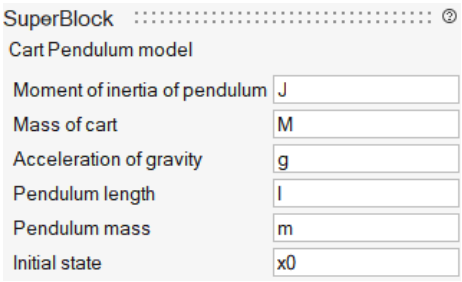
The following enhancements are available:

- The requirement to add volume components between components is resolved. Many components now have dead volumes at their hydraulic ports and can be connected directly to each other without the need for additional volume.
- An initialization issue with cylinder components has been fixed.

Code Generation

- Exposable Parameters: code generators (including FMU export) so far provided only two options: replace all model parameters with their numerical values in case of inline code generation, and the option to expose the parameters of all the simulation functions used, in case of simulation-function-based generation. It is now possible to choose the parameters to expose in the model. This is done by defining them as model parameters and taking advantage of the masking functionality.

For example, the following image shows a masked Super Block.

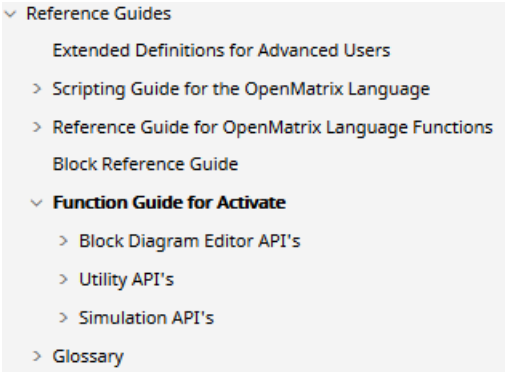


The parameters exposed in the generated code are J, M, g, l, m and x0. All non-structural parameters of blocks in the Base library (including FMUs) and Modelica components may depend on exposable parameters.

- FMU Exports: The Inline code generator can now be used for FMU export. The model may include an internal sample clock and Atomic super blocks*
- Inline Code Generator: The generated C code is improved. The variables are named so they can be traced back more easily to the blocks, and the application of basic operators and functions to large vectors and matrices can result in loop statements (no systematic unrolling of loops if the size exceeds a threshold). The code generation time has also been improved.*

Help Topics and Tutorials

- Documentation of more than 400 APIs has been added under ***Reference Guides*** > ***Function Guide for Activate***.



- Links to the Altair Community and Resource Library have been added to the help from ***Get Started*** > ***Online Learning***.
- Two new tutorials are available in the ***User Guide > Tutorials***: ***Define a Custom Block*** *and* ***Introduction to Modeling with Arduino***.

Resolved Issues

- Abnormal failure happens randomly when running certain MotionSolve co-simulation models repeatedly.*
- Modelica `zeroCrossings` block doesn't work.
- Missing Parameters in CombiTable (extrapolation).
- MotionSolve co-simulation model generating different results on Linux.*
- Undo not working for mixed Spice and explicit links.
- Undo link delete after deleting a super block port should not be allowed.
- Activate does not error out when MotionSolve license checkout fails.*
- The variable names in the FMU export should not contain escape characters.
- `Initial()` keyword doesn't work properly in Activate.
- Generation of C block and running simulation of Flux block was crashing Activate.*
- `vssImportSpiceLibrary` OML function crashing under Linux.
- Issues with paths in MotionSolve co-simulation across multiple machines.*
- Undo re-size in some blocks results in wrong position.
- Modelica LCCircuit demo had an incorrect block for the boolean pulse.
- FMU containing a Python Subroutine crashes.
- Inlined C-code generation was raising an issue with `Horner` block.
- C-code generator fails to compile due to 'nan' values.
- P project FMU export issue with Random block.
- Undo Port block creation causes crash.
- When exporting the super block of certain models that contain a single FMU, the exported FMU fails to simulate.
- Spice FMU export fails with error Spurious exception *Invalid_argument ("index out of bounds").*
- Activate batch doesn't check in license when using Altair One account.
- Activate batch doesn't work for Personal edition.
- CombiTable1D doesn't read MAT file when `vssInlineModelicaParameters(0)`
- Cannot read MAT files with `readRealMatrix` and `readMatrixSize` in Modelica compiler.
- Clocks were not treated correctly in FMI export
- Modelica Library AixLib not importing correctly.
- Importing a Modelica library should ask for a file, not only for a directory.
- Checkbox focus issue in Property Editor.
- Bad simulation settings crashes Spice.
- Setting for Flux install Path in Preferences does not take affect for co-simulation until restart.*
- Simulation failure with Spice FMU based on the value of the final time.
- Memory leak is observed after generating a model report for a large model.
- Specifying the result file name and its directory should not create 2 MTSF files.

- Press Ctrl+C during live printing in Python window crashes Activate.
- Ctrl-S does not save the model in a detached New Window.

***** ***Applies to Business Edition only.***

