ALTAIR
ONLY FORWARD

RELEASE NOTES

Altair® Twin Activate® 2024

ALTAIR © Altair Engineering, Inc. All Rights Reserved. / altair.com / Nasdaq: ALTR / Contact Us

New Features and Enhancements 2024

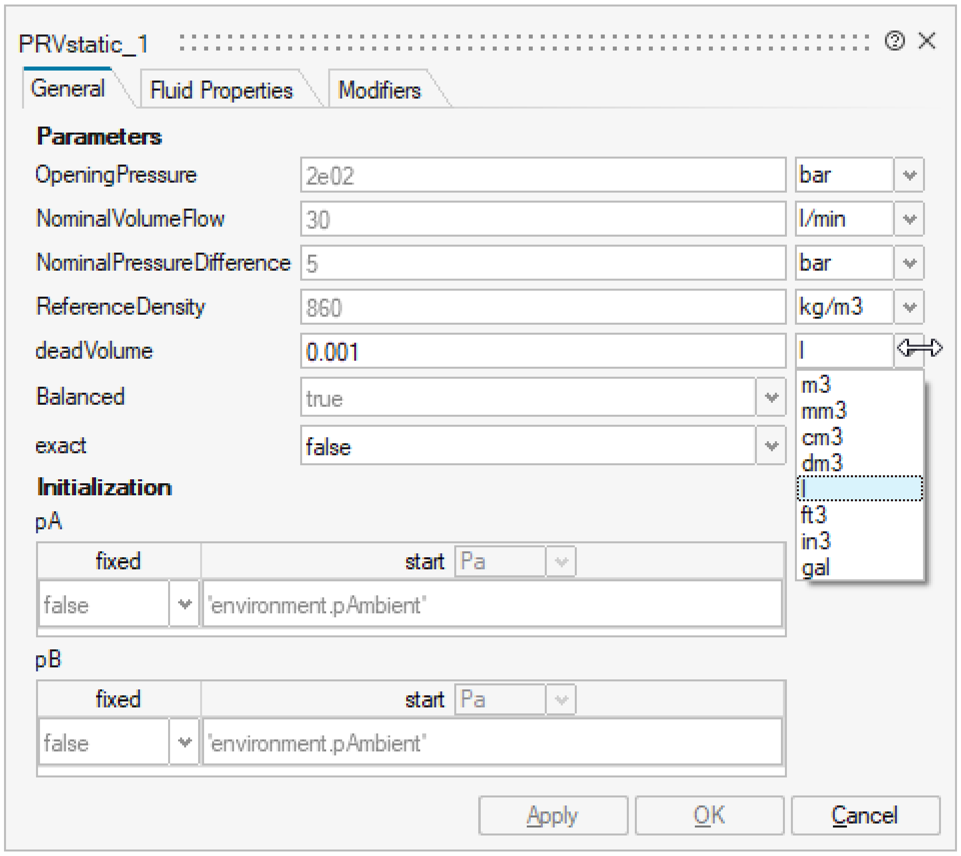
Release Highlights

User Interface

Select Units of Parameters and Initial Values in Component UI

Parameters and initial values are displayed with a combo box to select a unit other than the SI base unit. The value is automatically converted into the selected unit.

If the value input field contains an expression, or if you enter an expression, the unit switches to the basic SI unit and cannot be changed.



PRVstatic_1

General | Fluid Properties | Modifiers

Parameters

OpeningPressure	2e02	bar
NominalVolumeFlow	30	l/min
NominalPressureDifference	5	bar
ReferenceDensity	860	kg/m3
deadVolume	0.001	l
Balanced	true	
exact	false	

Unit options: m3, mm3, cm3, dm3, l, ft3, in3, gal

Initialization

pA

fixed	start Pa
false	"environment.pAmbient"

pB

fixed	start Pa
false	"environment.pAmbient"

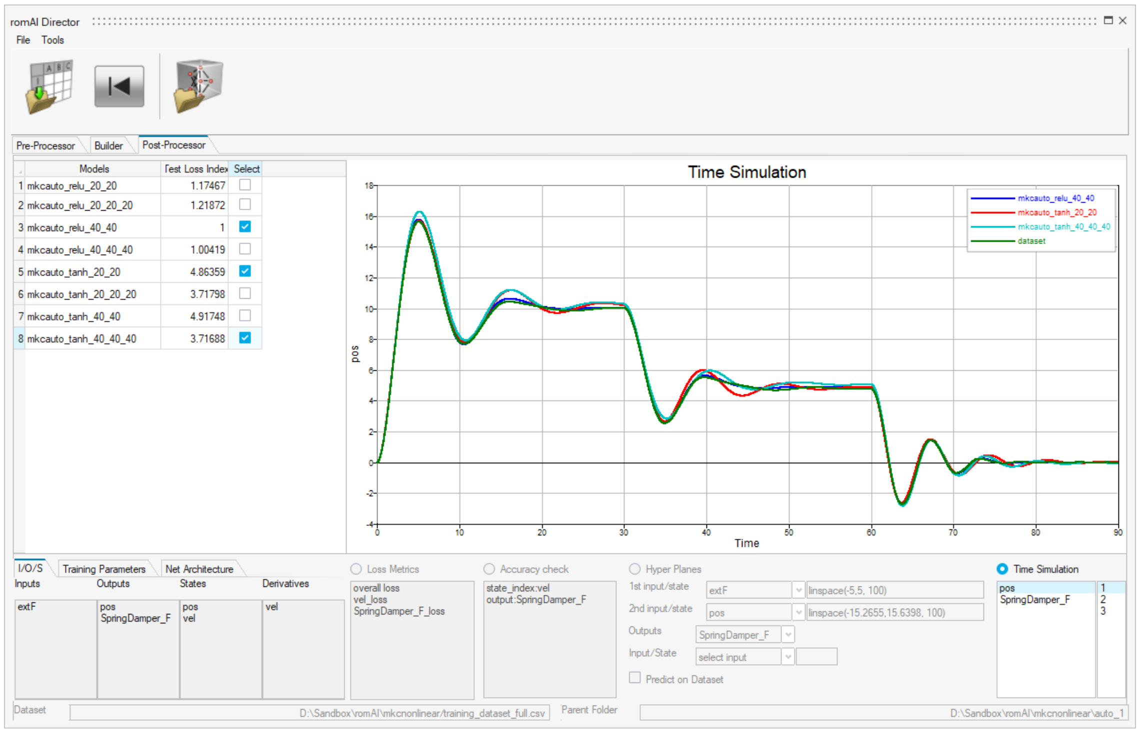
The solver of Twin Activate still uses SI base units. Display units are used only in UIs.

All libraries of Twin Activate are already prepared to use display units. Libraries created by a user must be imported again to support display units. This is an optional action. The library still works without re-import.

romAI Director*

Multi-Selection of Different romAI Models in Post Processing

The Automatic Exploration option is used to create multiple models for the same input data. In order to support the selection of the best model, the post/processor enables the selection of several models and the comparison of the time simulation results.



	Models	Test Loss Index	Select
1	mkcauto_relu_20_20	1.17467	☐
2	mkcauto_relu_20_20_20	1.21872	☐
3	mkcauto_relu_40_40	1	✓
4	mkcauto_relu_40_40_40	1.00419	☐
5	mkcauto_tanh_20_20	4.86359	✓
6	mkcauto_tanh_20_20_20	3.71798	☐
7	mkcauto_tanh_40_40	4.91748	☐
8	mkcauto_tanh_40_40_40	3.71688	✓

Inputs	Outputs	States	Derivatives
extF	pos SpringDamper_F	pos vel	vel

Loss Metrics: overall loss, vel_loss, SpringDamper_F_loss

Accuracy check: state_index:vel, output:SpringDamper_F

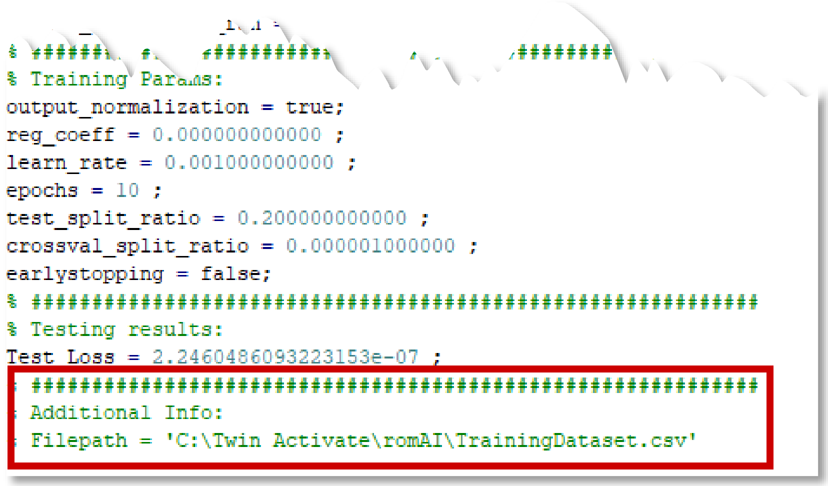
1st input/state: extF — linspace(-5,5, 100)
2nd input/state: pos — linspace(-15.2655,15.6398, 100)
Outputs: SpringDamper_F
Input/State: select input
☐ Predict on Dataset

Time Simulation: pos, SpringDamper_F — 1, 2, 3

Dataset: D:\Sandbox\romAI\mkcnonlinear/training_dataset_full.csv
Parent Folder: D:\Sandbox\romAI\mkcnonlinear\auto_1

Report of the Dataset Name/Location in the OML File Inside romAI

Information about the original dataset that was used to generate the model is provided in the OML file that is included in the romAI folder.

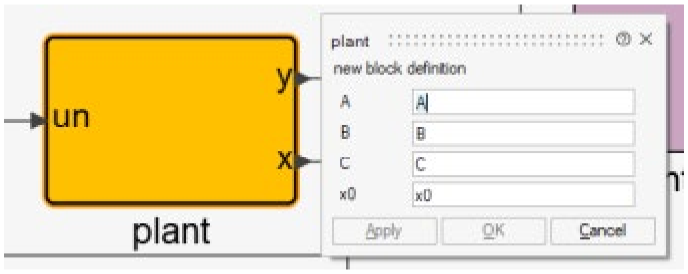


```
% Training Params:
output_normalization = true;
reg_coeff = 0.000000000000 ;
learn_rate = 0.001000000000 ;
epochs = 10 ;
test_split_ratio = 0.200000000000 ;
crossval_split_ratio = 0.000001000000 ;
earlystopping = false;
% ##############################################################
% Testing results:
Test Loss = 2.2460486093223153e-07 ;
% ##############################################################
% Additional Info:
% Filepath = 'C:\Twin Activate\romAI\TrainingDataset.csv'
```

Code Generation*

Python Target

- Python target now supports exposable parameters.



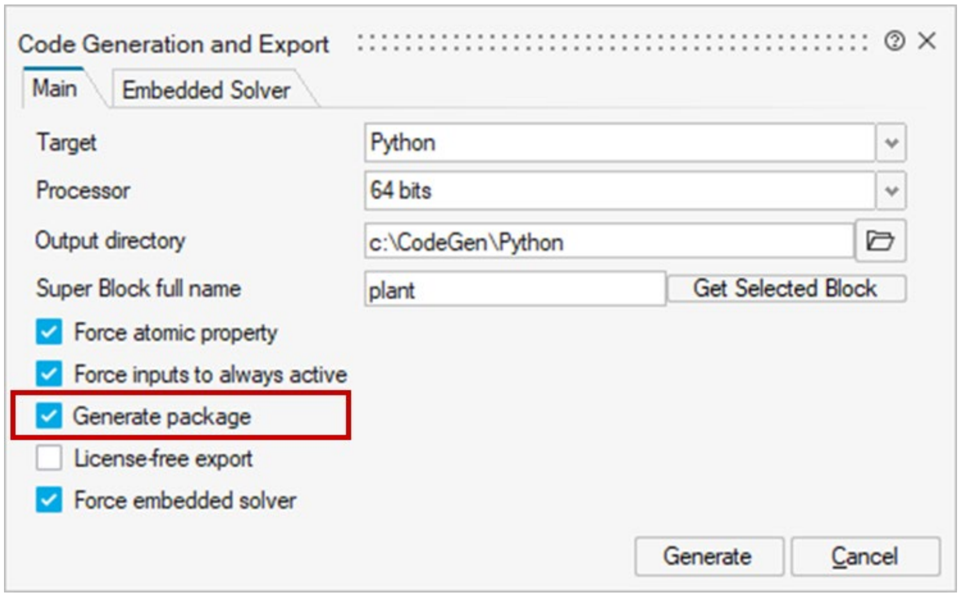
```
#!/usr/bin/env python

import numpy as np
import plant

def test():
    plant.init()
    A = np.array([1] * 9, dtype = np.double)
    B = np.array([1] * 3, dtype = np.double)
    C = np.array([1] * 6, dtype = np.double)
    x0 = np.array([1] * 3, dtype = np.double)
    plant.setparams(A, B, C, x0)
    input1 = np.array([1] * 1, dtype = np.double)
    inputs = [input1]
    plant.reset(inputs)
    t, outputs = plant.step(inputs, t = 1, dt = 0.1)
    print(t)
    print(outputs[0][0], outputs[0][1])
    print(outputs[1][0], outputs[1][1], outputs[1][2])
    plant.finish()

test()
```

- Python target now supports packages.


Code Generation and Export

Main | Embedded Solver

Target	Python
Processor	64 bits
Output directory	c:\CodeGen\Python
Super Block full name	plant

- ✓ Force atomic property
- ✓ Force inputs to always active
- ✓ Generate package
- ☐ License-free export
- ✓ Force embedded solver

The Generate package option generates a python package that builds a C/C++ extension via its *setup.py* script. To automatically compile the extension, install the package using the following:

```
import hwx.activate.apis as apis
apis.install_package('path to package')
```

After installation, you can run the test run 'path to package'/test/test.

Tutorials

Process and Visualize MQTT Data with Panopticon

One of Twin Activate's communication protocol options is MQTT, a well-known communication protocol used in IoT. It is possible to both publish and subscribe to any MQTT broker, local or external, very quickly and lightly.

This tutorial shows how to use Panopticon, a comprehensive data visualization and streaming analytics tool, as a client to subscribe to that broker, to process, and to visualize the data in real-time.

Create a Reduced-Order Model (ROM) of a Mass Spring Damper System Using Machine Learning

This tutorial demonstrates how to create a ROM of a dynamic system using romAI, starting with data and ending with the use of the romAI block in a simulation model.

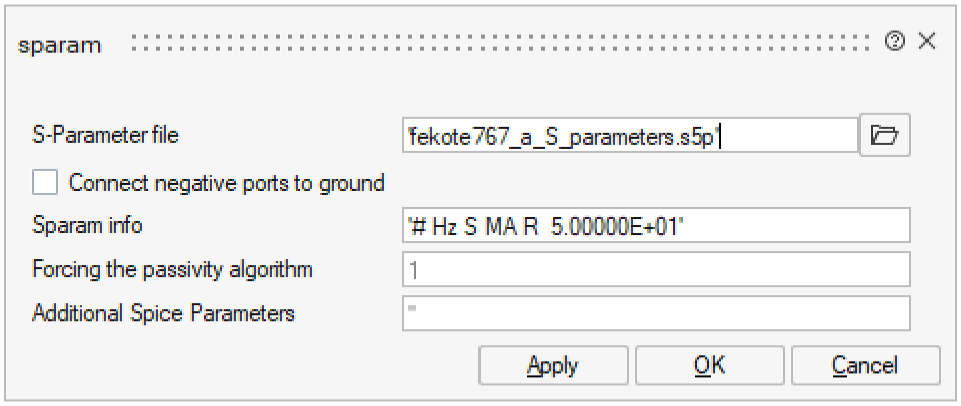
Create a GUI with UI Designer to Interact with a Simulation

This tutorial demonstrates the creation of a UI with UI Designer and the use in Twin Activate to change block parameters and run a simulation.

Libraries

Spice

- Library update to the latest version of HyperSpice
- Sparam block

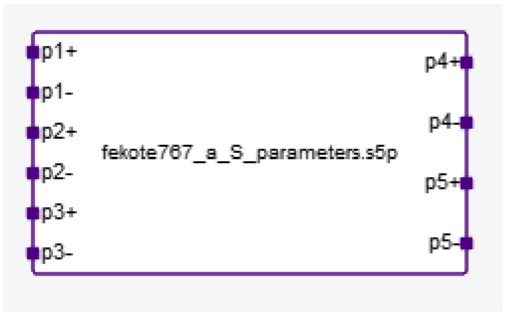


sparam

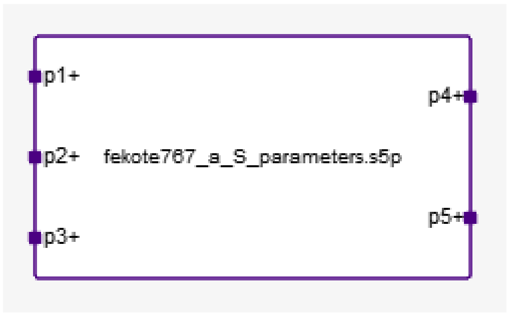
S-Parameter file	'fekote767_a_S_parameters.s5p'
☐ Connect negative ports to ground	
Sparam info	'# Hz S MA R 5.00000E+01'
Forcing the passivity algorithm	1
Additional Spice Parameters	''

- The number of ports resp. pins is set automatically depending on the file extension.

  snp > n ports, 2*n pins (pi1+, pi1, ..., pin+, pin-)



- Checkbox to connect pin- pins automatically to ground



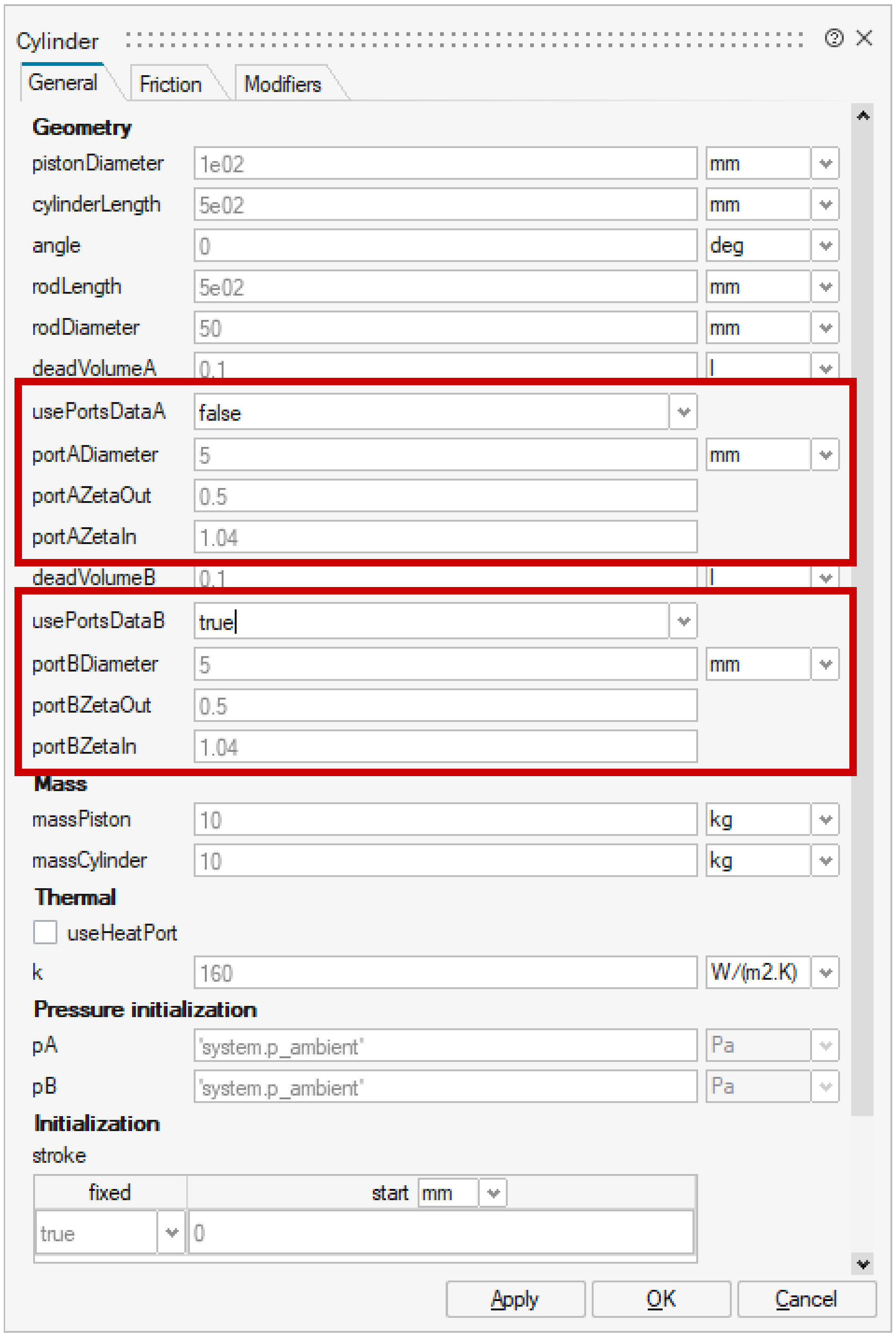
- Display of sparam info

Other Enhancements for Libraries

- Lookup1D, Lookup2D, Lookup3d, and Lookupnd support regular scale vectors.

AltairPneumatics

- Cylinder models extended to enable/disable port losses



Geometry		
pistonDiameter	1e02	mm
cylinderLength	5e02	mm
angle	0	deg
rodLength	5e02	mm
rodDiameter	50	mm
deadVolumeA	0.1	l
usePortsDataA	false	
portADiameter	5	mm
portAZetaOut	0.5	
portAZetaIn	1.04	
deadVolumeB	0.1	l
usePortsDataB	true	
portBDiameter	5	mm
portBZetaOut	0.5	
portBZetaIn	1.04	
Mass		
massPiston	10	kg
massCylinder	10	kg
Thermal		
☐ useHeatPort		
k	160	W/(m2.K)
Pressure initialization		
pA	'system.p_ambient'	Pa
pB	'system.p_ambient'	Pa
Initialization		
stroke		

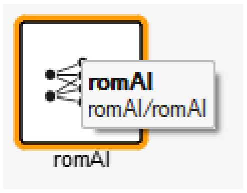
fixed	start mm
true	0

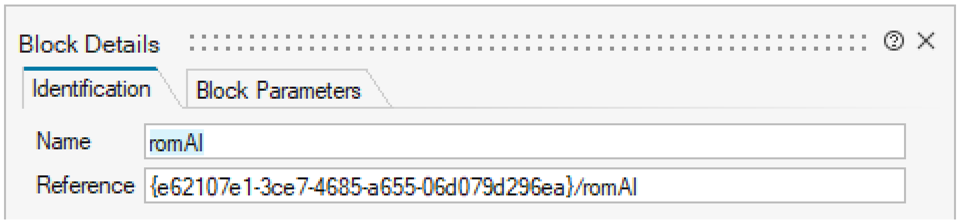
Enhancements

Enhancements for UI

- Project Browser now has an option to show only model names without blocks, super blocks, and annotations.
- Search in palettes is always a global search regardless of the currently selected branch.

- The block tool tip shows the library name instead of the library key (reference). The reference is still visible in the Block Details dialog.

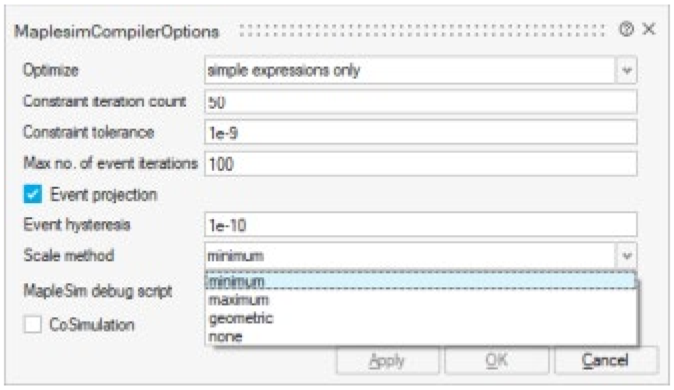




Name: romAI
Reference: {e62107e1-3ce7-4685-a655-06d079d296ea}/romAI

Enhancements for Modelica

- Scale method exposed in MaplesimCompilerOptions (Palettes/ModelicaExtras/Compiler Settings)



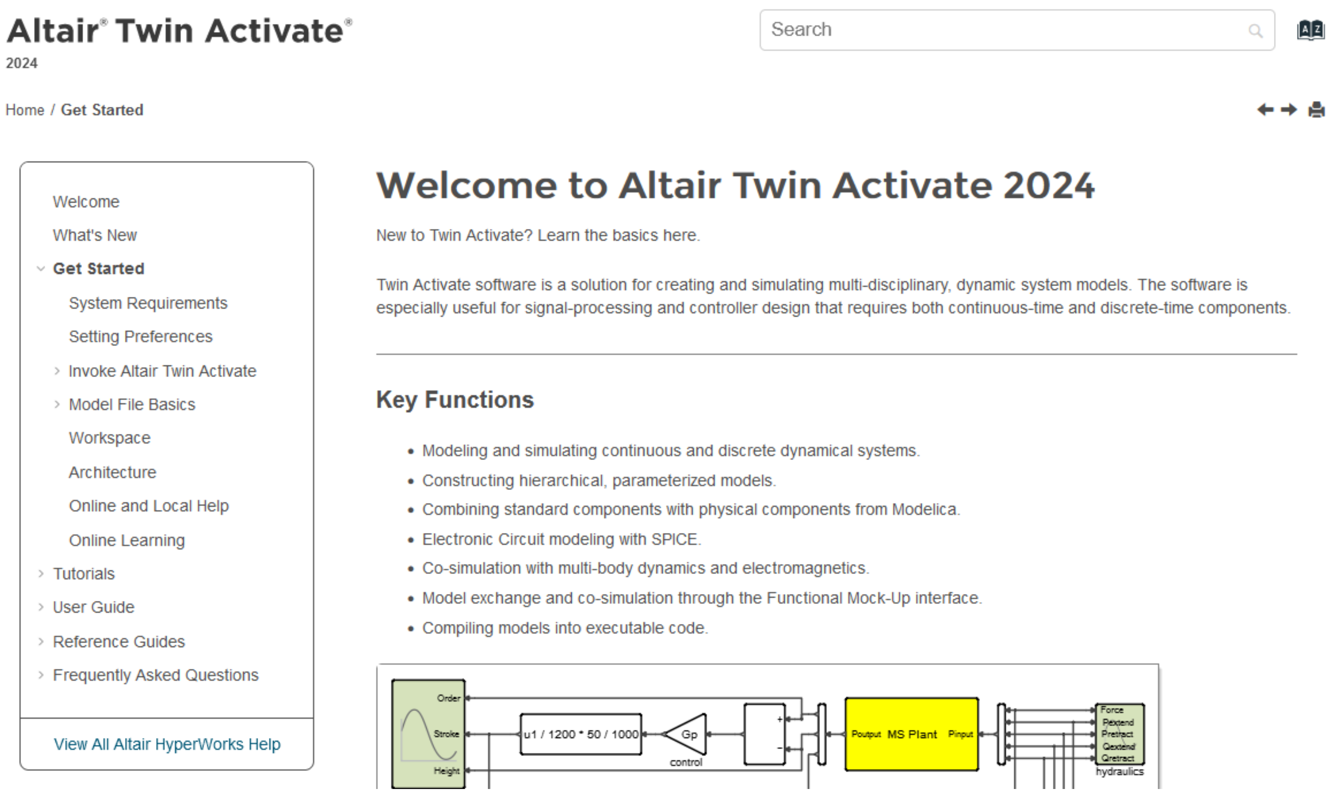
Optimize: simple expressions only
Constraint iteration count: 50
Constraint tolerance: 1e-9
Max no. of event iterations: 100
☑ Event projection
Event hysteresis: 1e-10
Scale method: minimum
MapleSim debug script
☐ CoSimulation

Scale method specifies the scaling method to be used when solving the problem. If not specified, the default is none. Correctly scaled variables are very helpful for convergence-based numerical algorithms, and sub-problem conditioning, so use of scaling for models containing variables with either very large or very small values allows for faster solution and greater accuracy.

- Memory dump inside the mocompiler.out is disabled.

Enhancements for Help

- The visual style of the help has been updated to a cleaner design.



Welcome
What's New
Get Started
System Requirements
Setting Preferences
Invoke Altair Twin Activate
Model File Basics
Workspace
Architecture
Online and Local Help
Online Learning
Tutorials
User Guide
Reference Guides
Frequently Asked Questions

View All Altair HyperWorks Help

Welcome to Altair Twin Activate 2024

New to Twin Activate? Learn the basics here.

Twin Activate software is a solution for creating and simulating multi-disciplinary, dynamic system models. The software is especially useful for signal-processing and controller design that requires both continuous-time and discrete-time components.

Key Functions

- Modeling and simulating continuous and discrete dynamical systems.
- Constructing hierarchical, parameterized models.
- Combining standard components with physical components from Modelica.
- Electronic Circuit modeling with SPICE.
- Co-simulation with multi-body dynamics and electromagnetics.
- Model exchange and co-simulation through the Functional Mock-Up interface.
- Compiling models into executable code.

Enhancements for OML Functions

New Functions

Serial Port

This library communicates between the server and client applications via serial port.

The library contains following command:

- **serialport**: Connects and communicates with serial port with the methods:
  - **read**: Reads data from a serial port object.
  - **write**: Writes data to a serial port object.
  - **close**: Closes serial port.

FTP

This library communicates between the server and client via file transfer protocol and a computer network.

The library contains the following commands:

- **ftp**: Create an FTP object to connect to a remote FTP server.

Supported file operation functions via FTP protocol are:

- **ascii**: Sets FTP connection to use ascii mode of transfer.
- **binary**: Sets FTP connection to use binary mode of transfer.
- **cd**: Change or get the current directory on the remote server over FTP connection.
- **close**: Close the FTP connection to the server.

- **delete**: Deletes a file over an FTP connection.
- **dir**: List the current directory or given directory of an FTP server.
- **mget**: Download files over FTP connection in the current working directory.
- **mkdir**: Create a new directory in a server over FTP connection.
- **mput**: Upload files to an FTP server.
- **rename**: Rename a file or directory on an FTP server.
- **rmdir**: Delete a directory of an FTP server.

Control Systems

- **rlocfind**: The root locus plot can be made interactive by using the mouse location to provide the input p and then display the outputs in a text box.
- [k, p] = rlocfind(sys, p)

Signal Processing

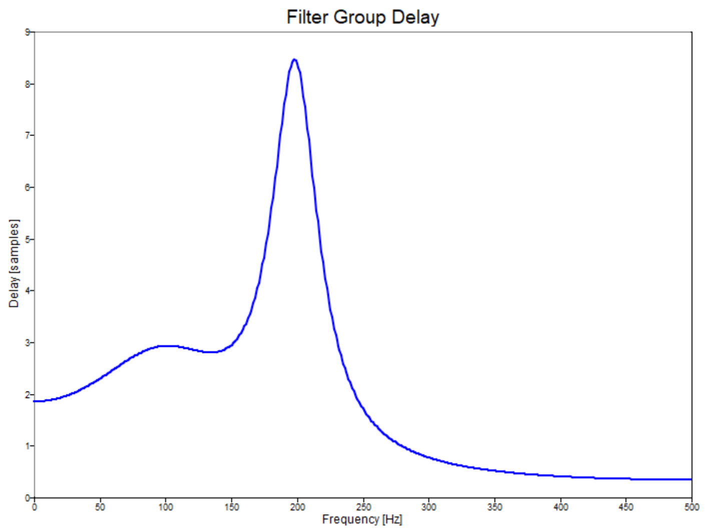
- **hilbert**: OML command that takes real signal and converts it to analytical signal using hilbert transformation
- **medfilt1**: Applies a moving median filter in one dimension.
- **grpdelay**: Compute digital filter group delay values.

```
fc = 200;

fs = 1000;

[b,a] = cheby1(4,1,fc/(fs/2));

grpdelay(b,a,[],fs);
```



Statistical Analysis

- **movmedian**: Computes moving median values with wide properties of endpoints.

  Supported end points are:

  - shrink: Windows that would extend beyond the matrix boundary shrink to contain only the existing data.

  - discard: Windows that would extend beyond the matrix boundary are discarded with the result that the output dimensions are reduced.
  - fill: Window elements that extend beyond the matrix boundary are filled with NaN values.
  - same: Window elements that extend beyond the matrix boundary are filled with the values of the elements on the matrix boundary.
  - periodic: Window elements that extend beyond the matrix boundary are filled by wrapping around the other end of each dimension.
  - number: Window elements that extend beyond the matrix boundary are filled with a specified numeric value.
- **randi**: Return random integers in the range of 1 to user-specified number.

```
    r = randi(idmax) --> returns a single random integer between 1:idmax

    r = randi(idmax,n) --> returns matrix of size n X n random integers between 1:idmax

    r = randi(idmax,n,m,...) --> returns matrix of size n X m random integers between 1:idmax

r = randi([idmin: idmax], ...) --> returns random integers between idmin:idmax
```

Resolved Issues

UI

- Manually modified port properties are not retained when copied and pasted.

OML

- arsm, grsm, moga functions are not available in Twin Activate.

** Applies to Commercial Edition only*