Release notes of Modelica standard library version 2.2.1

Version 2.2.1 is backward compatible to version 2.2.

In this version, **no** new libraries have been added. The following major improvements have been made:

- The **Documentation** of the Modelica standard library was considerably improved:
  In Dymola 6, the new feature was introduced to automatically add tables for class content and component interface definitions (parameters and connectors) to the info layer. For this reason, the corresponding (partial) tables previously present in the Modelica Standard Library have been removed. The new feature of Dymola 6 has the significant advantage that all tables are now guaranteed to be up-to-date.
  Additionally, the documentation has been improved by adding appropriate description texts to parameters, connector instances, function input and output arguments etc., in order that the automatically generated tables do not have empty entries. Also new users guides for sublibraries Rotational and SIunits have been added and the users guide on top level (Modelica.UsersGuide) has been improved.

- Initialization options have been added to the Modelica.Blocks.**Continuous** blocks (NoInit, SteadyState, InitialState, InitialOutput). If InitialOutput is selected, the block output is provided as initial condition. The states of the block are then initialized as close as possible to steady state. Furthermore, the Continuous.LimPID block has been significantly improved and much better documented.

- The Modelica.**Media** library has been significantly improved:
  New functions setState_pTX, setState_phX, setState_psX, setState_dTX have been added to PartialMedium to compute the independent medium variables (= state of medium) from p,T,X, or from p,h,X or from p,s,X or from d,T,X. Then functions are provided for all interesting medium variables to compute them from its medium state. All these functions are implemented in a robust way for all media (with a few exceptions, if the generic function does not make sense for a particular medium).

The following **new components** have been added to **existing** libraries:

Modelica.Blocks.Examples.	
PID_Controller	Example to demonstrate the usage of the Blocks.Continuous.LimPID block.
Modelica.Blocks.Math.	
UnitConversions.*	New package that provides blocks for unit conversions. UnitConversions.ConvertAllBlocks allows to select all available conversions from a menu.
Modelica.Electrical.Machines.BasicMachines.SynchronousInductionMachines.	
SM_ElectricalExcitedDamperCage	Electrical excited synchronous induction machine with damper cage
Modelica.Electrical.Machines.BasicMachines.Components.	
ElectricalExcitation	Electrical excitation for electrical excited synchronous induction machines
DamperCage	Unsymmetrical damper cage for electrical excited synchronous induction machines. At least the user has to specify the dampers resistance and stray inductance in d-axis; if he omits the parameters of the q-axis, the same values as for the d.axis are used, assuming a symmetrical damper.
Modelica.Electrical.Machines.Examples.	
SMEE_Gen	Test example 7: ElectricalExcitedSynchronousInductionMachine as Generator
Utilities.TerminalBox	Terminal box for three-phase induction machines to choose either star (wye) ? or delta ? connection
Modelica.Math.Matrices.	
equalityLeastSquares	Solve a linear equality constrained least squares problem: min|A*x-a|^2 subject to B*x=b
Modelica.Mechanics.MultiBody.	
Parts.PointMass	Point mass, i.e., body where inertia tensor is neglected.
Interfaces.FlangeWithBearing	Connector consisting of 1-dim. rotational flange and its 3-dim. bearing frame.
Interfaces.FlangeWithBearingAdaptor	Adaptor to allow direct connections to the sub-connectors of FlangeWithBearing.
Types.SpecularCoefficient	New type to define a specular coefficient.
Types.ShapeExtra	New type to define the extra data for visual shape objects and to have a central place for the documentation of this data.
Modelica.Mechanics.MultiBody.Examples.Elementary	
PointGravityWithPointMasses	Example of two point masses in a central gravity field.
Modelica.Mechanics.Rotational.	
UsersGuide	A users guide has been added by using the documentation previously present in the package documentation of Rotational.
Sensors.PowerSensor	New component to measure the energy flow between two connectors of the Rotational library.
Modelica.Mechanics.Translational.	
Speed	New component to move a translational flange according to a reference speed

Modelica.Media.Interfaces.PartialMedium.	
specificEnthalpy_pTX	New function to compute specific enthalpy from pressure, temperature and mass fractions.
temperature_phX	New function to compute temperature from pressure, specific enthalpy, and mass fractions.
Modelica.Icons.	
SignalBus	Icon for signal bus
SignalSubBus	Icon for signal sub-bus
Modelica.SIunits.	
UsersGuide	A users guide has been added that describes unit handling.
Resistance Conductance	Attribute 'min=0' removed from these types.
Modelica.Thermal.FluidHeatFlow.	
Components.Valve	Simple controlled valve with either linear or exponential characteristic.
Sources. IdealPump	Simple ideal pump (resp. fan) dependent on the shaft?s speed; pressure increase versus volume flow is defined as a linear function. Torque * Speed = Pressure invrease * Volume flow (without losses).
Examples.PumpAndValve	Test example for valves.
Examples.PumpDropOut	Drop out of 1 pump to test behavior of semiLinear.
Examples.ParallelPumpDropOut	Drop out of 2 parallel pumps to test behavior of semiLinear.
Examples.OneMass	Cooling of 1 hot mass to test behavior of semiLinear.
Examples.TwoMass	Cooling of 2 hot masses to test behavior of semiLinear.

The following **components** have been improved:

Modelica.	
UsersGuide	Users guide and package description of Modelica Standard Library improved.
Modelica.Blocks.Interfaces.	
RealInput BooleanInput IntegerInput	When dragging one of these connectors the width and height is a factor of 2 larger as a standard icon. Previously, these connectors have been dragged and then manually enlarged by a factor of 2 in the Modelica standard library.
Modelica.Blocks.	
Continuous.*	Initialization options added to all blocks (NoInit, SteadyState, InitialState, InitialOutput). New parameter limitsAtInit to switch off the limits of LimIntegrator or LimPID during initialization
Continuous.LimPID	Option to select P, PI, PD, PID controller. Documentation significantly improved.
Nonlinear.Limiter Nonlinear.VariableLimiter Nonlinear.Deadzone	New parameter limitsAtInit/deadZoneAtInit to switch off the limits or the dead zone during initialization
Modelica.Electrical.Analog.	
Sources	Icon improved (+/- added to voltage sources, arrow added to current sources).
Modelica.Electrical.Analog.Semiconductors.	
Diode	smooth() operator included to improve numerics.
Modelica.Electrical.Machines.BasicMachines.SynchronousInductionMachines.	
SM_PermanentMagnetDamperCage SM_ElectricalExcitedDamperCage SM_ReluctanceRotorDamperCage	The user can choose \"DamperCage = false\" (default: true) to remove all equations for the damper cage from the model.
Modelica.Electrical.Machines.BasicMachines.AsynchronousInductionMachines.	
AIM_SlipRing	Easier parameterization: if the user selects \"useTrunsRatio = false\" (default: true, this is the same behavior as before), parameter TurnsRatio is calculated internally from Nominal stator voltage and Locked-rotor voltage.
Modelica.Math.Matrices.	
leastSquares	The A matrix in the least squares problem might be rank deficient. Previously, it was required that A has full rank.
Modelica.Mechanics.MultiBody.	
all models	• All components with animation information have a new variable **specularCoefficient** to define the reflection of ambient light. The default value is world.defaultSpecularCoefficient which has a default of 0.7. By changing world.defaultSpecularCoefficient, the specularCoefficient of all components is changed that are not explicitly set differently. Since specularCoefficient is a variable (and no parameter), it can be changed during simulation. Since annotation(Dialog) is set, this variable still appears in the parameter menus.

	Previously, a constant specularCoefficient of 0.7 was used for all components. • Variable **color** of all components is no longer a parameter but an input variable. Also all parameters in package **Visualizers**, with the exception of **shapeType** are no longer parameters but defined as input variables with annotation(Dialog). As a result, all these variables appear still in parameter menus, but they can be changed during simulation (e.g., color might be used to display the temperature of a part). • All menus have been changed to follow the Modelica 2.2 annotations \"Dialog, group, tab, enable\" (previously, a non-standard Dymola definition for menus was used). Also, the \"enable\" annotation is used in all menus to disable input fields if the input would be ignored. • All visual shapes are now defined with conditional declarations (to remove them, if animation is switched off). Previously, these (protected) objects have been defined by arrays with dimension 0 or 1.
Frames.resolveRelative	The derivative of this function added as function and defined via an annotation. In certain situations, tools had previously difficulties to differentiate the inlined function automatically.
Forces.*	The scaling factors N_to_m and Nm_to_m have no longer a default value of 1000 but a default value of world.defaultN_to_m (=1000) and world.defaultNm_to_m (=1000). This allows to change the scaling factors for all forces and torques in the world object.
Interfaces.Frame.a Interfaces.Frame.b Interfaces.Frame_resolve	The Frame connectors are now centered around the origin to ease the usage. The shape was changed, such that the icon is a factor of 1.6 larger as a standard icon (previously, the icon had a standard size when dragged and then the icon was manually enlarged by a factor of 1.5 in the y-direction in the MultiBody library; the height of 16 allows easy positioning on the standard grid size of 2). The double line width of the border in icon and diagram layer was changed to a single line width and when making a connection the connection line is dark grey and no longer black which looks better.
Joints.Assemblies.*	When dragging an assembly joint, the icon is a factor of 2 larger as a standard icon. Icon texts and connectors have a standard size in this enlarged icon (and are not a factor of 2 larger as previously).
Types.*	All types have a corresponding icon now to visualize the content in the package browser (previously, the types did not have an icon).
Modelica.Mechanics.Rotational.	
Inertia	Initialization and state selection added.
SpringDamper	Initialization and state selection added.
Move	New implementation based solely on Modelica 2.2 language (previously, the Dymola specific constrain(..) function was used).
Modelica.Mechanics.Translational.	
Move	New implementation based solely on Modelica 2.2 language (previously, the Dymola specific constrain(..) function was used).
Modelica.Thermal.FluidHeatFlow.Interfaces.Partials.	
SimpleFriction	Calculates friction losses from pressure drop and volume flow.
Modelica.Thermal.FluidHeatFlow.Components.	
IsolatedPipe HeatedPipe	Added geodetic height as a source of pressure change; feeds friction losses as calculated by simple friction to the energy balance of the medium.
Modelica.Media.Interfaces.PartialMedium.FluidConstants.	
HCRIT0	Critical specifc enthalpy of the fundamental equation (base formulation of the fluid medium model).
SCRIT0	Critical specifc entropy of the fundamental equation (base formulation of the fluid medium model).
deltah	Enthalpy offset (default: 0) between the specific enthalpy of the fluid model and the user-visible specific enthalpy in the model: deltah = h_model - h_fundamentalEquation.
deltas	Entropy offset (default: 0) between the specific entropy of the fluid model and the user-visible specific entropy in the model: deltas = s_model - s_fundamentalEquation.
T_default	Default value for temperature of medium (for initialization)
h_default	Default value for specific enthalpy of medium (for initialization)
p_default	Default value for pressure of medium (for initialization)
X_default	Default value for mass fractions of medium (for initialization)

The following **errors** have been fixed:

Modelica.Blocks.Tables.	
CombiTable1D CombiTable1Ds CombiTable2D	Parameter \"tableOnFile\" determines now whether a table is read from file or used from parameter \"table\". Previously, if \"fileName\" was not \"NoName\", a table was always read from file \"fileName\", independently of the setting of \"tableOnFile\". This has been corrected. Furthermore, the initialization of a table is now performed in a when-clause and no longer in a parameter declaration, because some tools evaluate the parameter declaration in some situation more than once and

	then the table is unnecessarily read several times (and occupies also more memory).
Modelica.Blocks.Sources.	
CombiTimeTable	Same bug fix/improvement as for the tables from Modelica.Blocks.Tables as outlined above.
Modelica.Electrical.Analog.Semiconductors.	
PMOS NMOS HeatingPMOS HeatingNMOS	The Drain-Source-Resistance RDS had actually a resistance of RDS/v, with v=Beta*(W+dW)/(L+dL). The correct formula is without the divison by \"v\". This has now been corrected. This bug fix should not have an essential effect in most applications. In the default case (Beta=1e-5), the Drain-Source-Resistance was a factor of 1e5 too large and had in the default case the wrong value 1e12, although it should have the value 1e7. The effect was that this resistance had practically no effect.
Modelica.Media.IdealGases.Common.SingleGasNasa.	
dynamicViscosityLowPressure	Viscosity and thermal conductivity (which needs viscosity as input) were computed wrong for polar gases and gas mixtures (i.e. if dipole moment not 0.0). This has been fixed in version 2.2.1.
Modelica.Utilities.Streams.	
readLine	Depending on the C-implementation, the stream was not correctly closed. This has been corrected by adding a \"Streams.close(..)\" after reading the file content.