

Release Notes of Modelica Standard Library

Version 2.1 (October 1, 2004)

This is a major change with respect to previous versions of the Modelica Standard Library, because **many new libraries** and components are included and because the input/output blocks (Modelica.Blocks) have been considerably simplified:

- An input/output connector is defined **without** a hierarchy (this is possible due to new features of the Modelica language). For example, the input signal of a block "FirstOrder" was previously accessed as "FirstOrder.inPort.signal[1]". Now it is accessed as "FirstOrder.u". This simplifies the understanding and usage especially for beginners.
- De-vectorized the **Modelica.Blocks** library. All blocks in the Modelica.Blocks library are now scalar blocks. As a result, the parameters of the Blocks are scalars and no vectors any more. For example, a parameter "amplitude" that might had a value of "{1}" previously, has now a value of "1". This simplifies the understanding and usage especially for beginners. If a vector of blocks is needed, this can be easily accomplished by adding a dimension to the instance. For example "Constant const[3](k={1,2,3}" defines three Constant blocks. An additional advantage of the new approach is that the implementation of Modelica.Blocks is much simpler and is easier to understand.

The discussed changes of Modelica.Blocks are not backward compatible. A script to **automatically** convert models to this new version is provided. There might be rare cases, where this script does not convert. In this case models have to be manually converted. In any case you should make a back-up copy of your model before automatic conversion is performed.

The following **new libraries** have been added:

Modelica.Electrical.Digital	Digital electrical components based on 2-,3-,4-, and 9-valued logic according to the VHDL standard
Modelica.Electrical.Machines	Asynchronous, synchronous and DC motor and generator models
Modelica.Math.Matrices	Functions operating on matrices such as solve() ($A*x=b$), leastSquares(), norm(), LU(), QR(), eigenValues(), singularValues(), exp(), ...
Modelica.StateGraph	Modeling of discrete event and reactive systems in a convenient way using hierarchical state machines and Modelica as action language . It is based on JGraphChart and Grafcet and has a similar modeling power as StateCharts. It avoids deficiencies of usually used action languages. This library makes the ModelicaAdditions.PetriNets library obsolete.
Modelica.Utilities.Files	Functions to operate on files and directories (copy, move, remove files etc.)
Modelica.Utilities.Streams	Read from files and write to files (print, readLine, readFile, error, ...)
Modelica.Utilities.Strings	Operations on strings (substring, find, replace, sort, scanToken, ...)
Modelica.Utilities.System	Get/set current directory, get/set environment variable, execute shell command, etc.

The following existing libraries outside of the Modelica standard library have been improved and added as **new libraries** (models using the previous libraries are automatically converted to the new sublibraries inside package Modelica):

Modelica.Blocks.Discrete	Discrete input/output blocks with fixed sample period (from ModelicaAdditions.Blocks.Discrete)
Modelica.Blocks.Logical	Logical components with Boolean input and output signals (from ModelicaAdditions.Blocks.Logical)
Modelica.Blocks.Nonlinear	Discontinuous or non-differentiable algebraic control blocks such as variable limiter, fixed, variable and Pade delay etc. (from ModelicaAdditions.Blocks.Nonlinear)
Modelica.Blocks.Routing	Blocks to combine and extract signals, such as multiplexer (from ModelicaAdditions.Blocks.Multiplexer)
Modelica.Blocks.Tables	One and two-dimensional interpolation in tables. CombiTimeTable is available in Modelica.Blocks.Sources (from ModelicaAdditions.Tables)
Modelica.Mechanics.MultiBody	Components to model the movement of 3-dimensional mechanical systems. Contains body, joint, force and sensor components, analytic handling of kinematic loops, force elements with mass, series/parallel connection of 3D force elements etc. (from MultiBody 1.0 where the new signal connectors are used; makes the ModelicaAdditions.MultiBody library obsolete)

As a result, the ModelicaAdditions library is obsolete, because all components are either included in the Modelica library or are replaced by much more powerful libraries (MultiBody, StateGraph).

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

Modelica.Blocks.Logical.	
Pre	$y = \text{pre}(u)$: Breaks algebraic loops by an infinitesimal small time delay (event iteration continues until $u = \text{pre}(u)$)
Edge	$y = \text{edge}(u)$: Output y is true, if the input u has a rising edge
FallingEdge	$y = \text{edge}(\text{not } u)$: Output y is true, if the input u has a falling edge
Change	$y = \text{change}(u)$: Output y is true, if the input u has a rising or falling edge
GreaterEqual	Output y is true, if input $u1$ is greater or equal as input $u2$
Less	Output y is true, if input $u1$ is less as input $u2$
LessEqual	Output y is true, if input $u1$ is less or equal as input $u2$
Timer	Timer measuring the time from the time instant where the Boolean input became true
Modelica.Blocks.Math.	
BooleanToReal	Convert Boolean to Real signal
BooleanToInteger	Convert Boolean to Integer signal
RealToBoolean	Convert Real to Boolean signal
IntegerToBoolean	Convert Integer to Boolean signal
Modelica.Blocks.Sources.	
RealExpression	Set output signal to a time varying Real expression
IntegerExpression	Set output signal to a time varying Integer expression
BooleanExpression	Set output signal to a time varying Boolean expression
BooleanTable	Generate a Boolean output signal based on a vector of time instants
Modelica.Mechanics.Rotational.	
LinearSpeedDependentTorque	Linear dependency of torque versus speed (acts as load torque)
QuadraticSpeedDependentTorque	Quadratic dependency of torque versus speed (acts as load torque)
ConstantTorque	Constant torque, not dependent on speed (acts as load torque)
ConstantSpeed	Constant speed, not dependent on torque (acts as load torque)
TorqueStep	Constant torque, not dependent on speed (acts as load torque)

The following **bugs** have been **corrected**:

Modelica.Mechanics.MultiBody.Forces.	
LineForceWithMass Spring	If mass of the line force or spring component is not zero, the mass was (implicitly) treated as "mass*mass" instead of as "mass"
Modelica.Mechanics.Rotational.	
Speed	If parameter exact= false , the filter was wrong (position was filtered and not the speed input signal)

Other changes:

- All connectors are now smaller in the diagram layer. This gives a nicer layout when connectors and components are used together in a diagram
- Default instance names are defined for all connectors, according to a new annotation introduced in Modelica 2.1. For example, when dragging connector "Flange_a" from the Rotational library to the diagram layer, the default connector instance name is "flange_a" and not "Flange_a1".
- The Modelica.Mechanics.Rotational connectors are changed from a square to a circle
- The Modelica.Mechanics.Translational connectors are changed from a green to a dark green color in order that connection lines can be better seen, especially when printed.
- The Modelica.Blocks connectors for Real signals are changed from blue to dark blue in order to distinguish them from electrical signals.

Statistics

	<i>number of models + blocks</i>	<i>number of functions</i>
Modelica 1.6	296	41
Modelica 2.1	576	206