

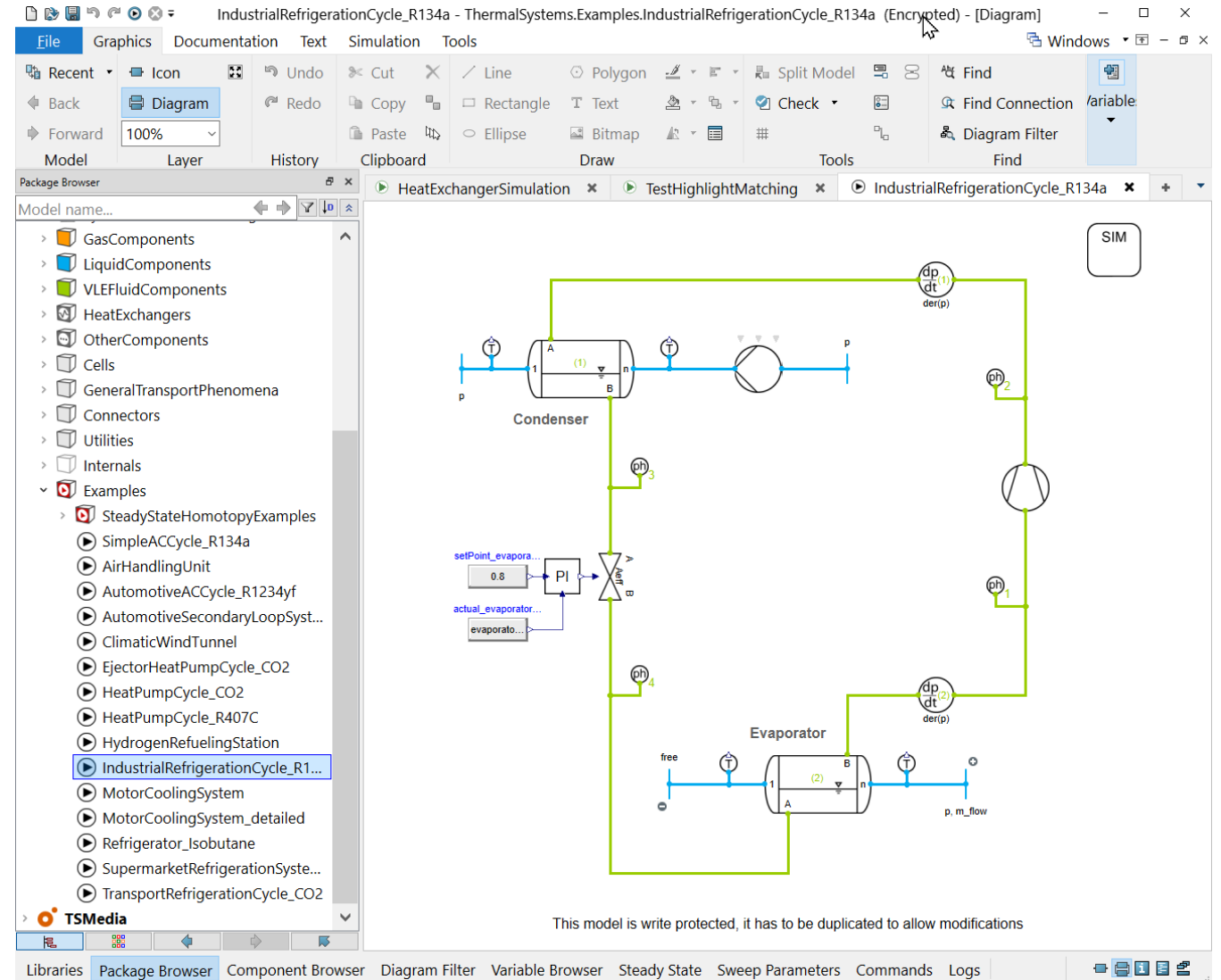


Neuigkeiten in Dymola 2024x Refresh 1

Matthias Schäfer

Agenda

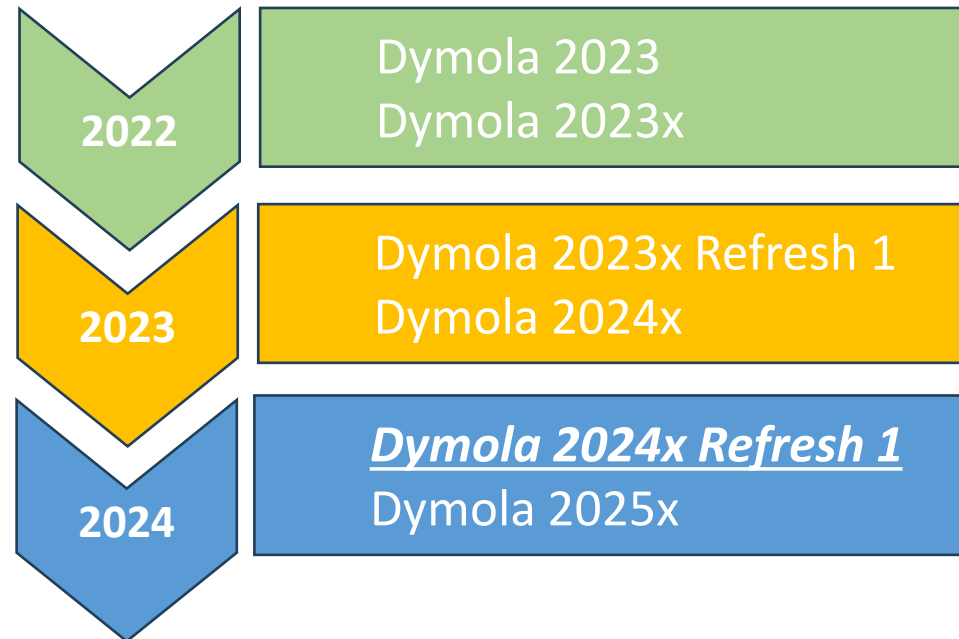
1. Dymola 2024x Refresh 1
 1. Modellierung
 2. Simulation
 3. Graphische Nutzeroberfläche
 4. Versionierung & Skripting
 5. Lizenzierung
2. Diskussion & Fragen



Angepasster Release-Cycle

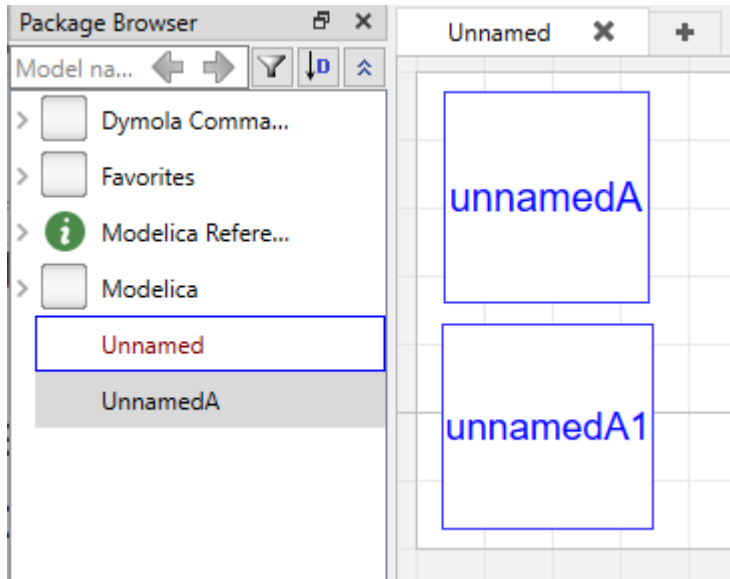
NEW RELEASE LIFECYCLE

- One major release per year (autumn)
 - Dymola 2023x
 - Dymola 2024x
 - And so on...
- One maintenance release per year (spring)
 - Dymola 2023x Refresh 1
 - Dymola 2024x Refresh 1



Modellierung

Default Nummerierung von neuen Modellen



Wenn man mit File → New → Model ein neues Modell erstellt, ist der default-Name nun UnnamedA, UnnamedB usw.

Neue Instanzen bekommen weiterhin das Suffix 1, 2 usw.

Dadurch lassen sich Klassen und Instanz-namen besser auseinanderhalten:

Bislang:



Ableitung unstetiger Funktionen

Mit dem Flag

Advanced.Translation.ImpulseWhenDifferentiatingDiscontinuity = true

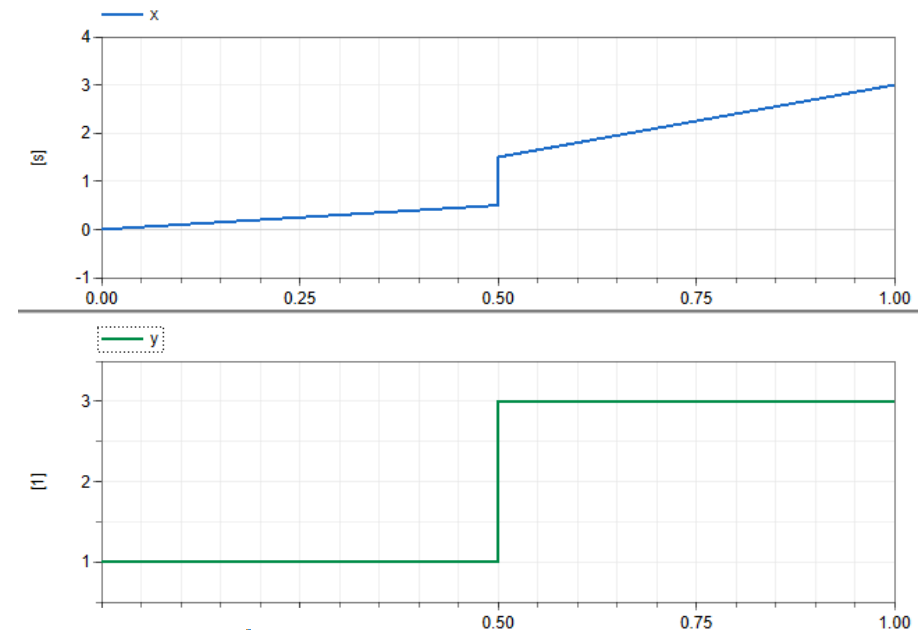
können unstetige Funktionen einmal abgeleitet werden

```
model DiscontinuousDerivative
  Real x= if time <0.5 then time else 3*time;
  Real y;

  equation
    y = der(x);

end DiscontinuousDerivative;
```

```
Introducing impulse to handle differentiated expression that was not continuous:
(if time < 0.5 then time else 3*time)
Value jumped from 0.5 to 1.5.
```



Ohne diesen Flag (default =false) und in früheren Dymola Versionen tritt ein Model error („Value jumped“) auf.
Diese Option besteht nur für Dymola-Modelle (keine FMUs, ...)

Dynamische Min/Max Grenzen

- Min/Max Grenzen für Variablen müssen nun keine festen Werte mehr sein

- Bislang:

```
Real x(min=0, max=1);
```

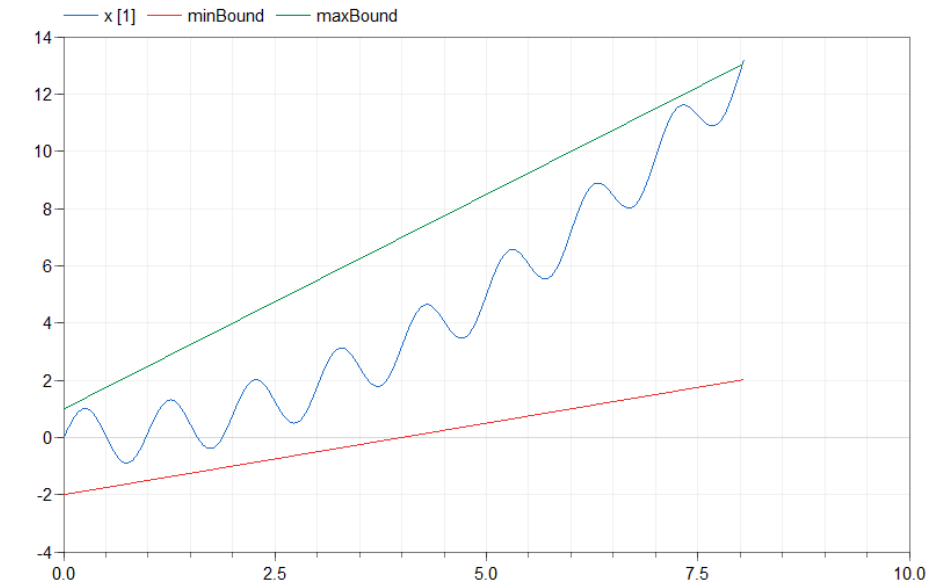
- Neu:

```
parameter Real k = 5;
```

```
parameter Real m = k/2;
```

```
Real x(min = myFunction(k), max = m);
```

- Das ermöglicht zum Beispiel zu Überprüfen, ob Signale innerhalb einer dynamischen Bandbreite liegen:



```
Integration started at T = 0 using integration method DASSL
(DAE multi-step solver (dassl/dasslrt of Petzold modified by Dassault Systeme:
Error: The following error was detected at time: 8.040000000000001
Value for variable, with min/max, is: 13.17700989
Min: 2.01
Max: 13.03
Failed condition: x <= maxBound
Integration terminated before reaching "StopTime" at T = 8.04
```

```
Advanced.AssertAllInsideMinMax = true;
```

Simulation

Neuer Solver: IDA

Das Solver-Paket von Sundials wurde aktualisiert (Version 6.4.1)
Damit steht in Dymola der neue Solver **Ida** zur Verfügung:

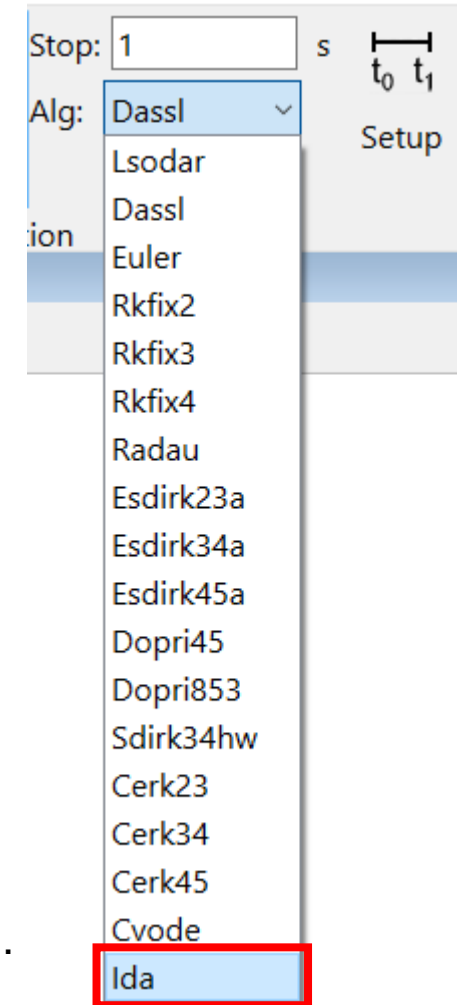
- moderner BDF-Solver
- erlaubt DAE- und Sparse- Integration
- kann in CoSim-FMUs verwendet werden

Empfehlung:

- Dassl bleibt die erste Wahl (Dymola's default)
- Wenn am Solver gedreht werden muss, sollte **Ida** auf jeden Fall in Erwägung gezogen werden



Mit dem Update von Sundials wurde auch der Ccode Solver aktualisiert.
Die alte Version ist nicht mehr verfügbar.



Vererbung der experiment annotation

Die experiment annotation wird nun aus extended Modellen übernommen

Beispiel:

```

model UnnamedA
  extends Unnamed;
  annotation(experiment(StopTime = 5, Tolerance = 1e-5));
end UnnamedA;

model Unnamed
  Real x(start=1);
  equation
    der(x)=-1*x;
  annotation(experiment(StopTime=10, __Dymola_Algorithm="Cvode"));
end Unnamed;
  
```

Ergebnis mit Dymola 2024x Refresh 1

```

Model: UnnamedA
Integration started at 0 using integration method:
cvode from sundials

Integration terminated successfully at T = 5
CPU-time for integration : 0.013 seconds
  
```

Logs Syntax Translation Simulation Version

Ergebnis mit Dymola 2024x

```

Integration started at T = 0 using integration method DASSL
(DAE multi-step solver (dassl/dasslrt of Petzold modified by

Integration terminated successfully at T = 5
  
```

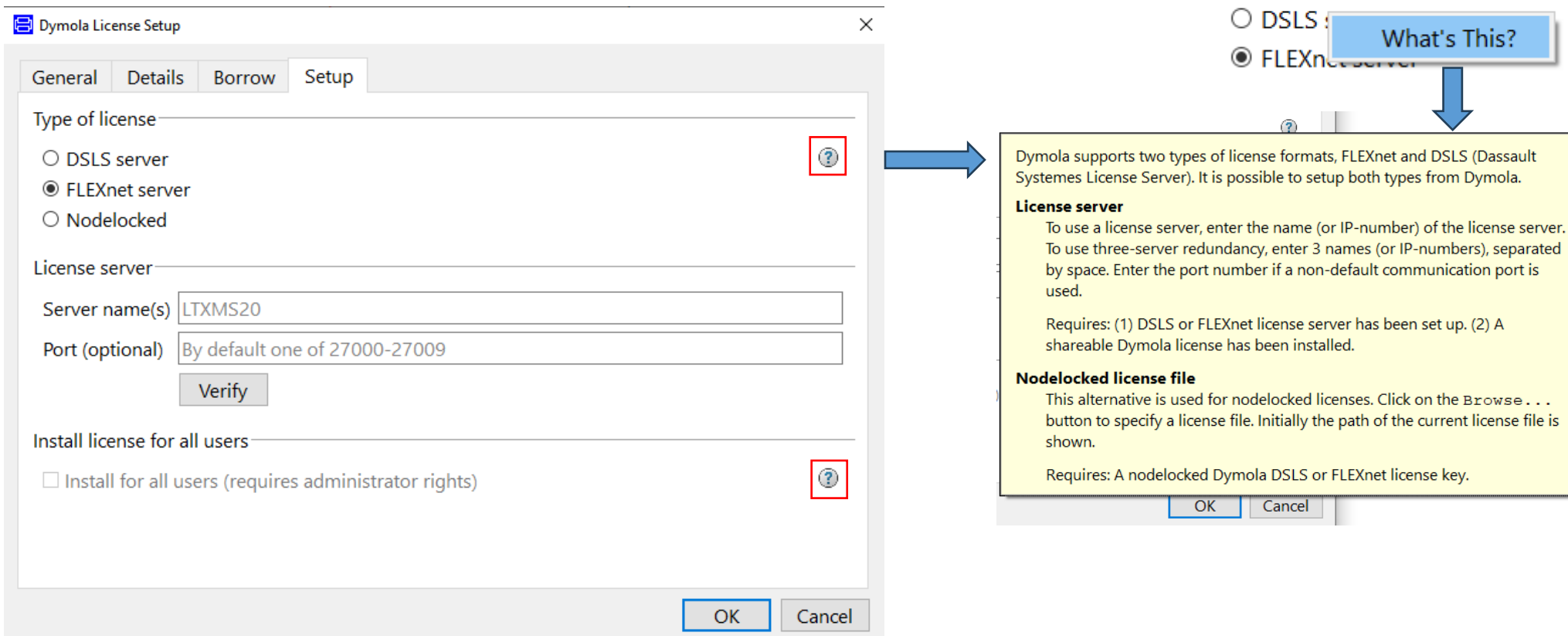
Logs Syntax Translation Simulation Version

Bei mehreren widersprüchlichen extend statements wird die erste verwendet
 experiment annotations aus Submodellen werden weiterhin nicht übernommen

Graphische Nutzeroberfläche

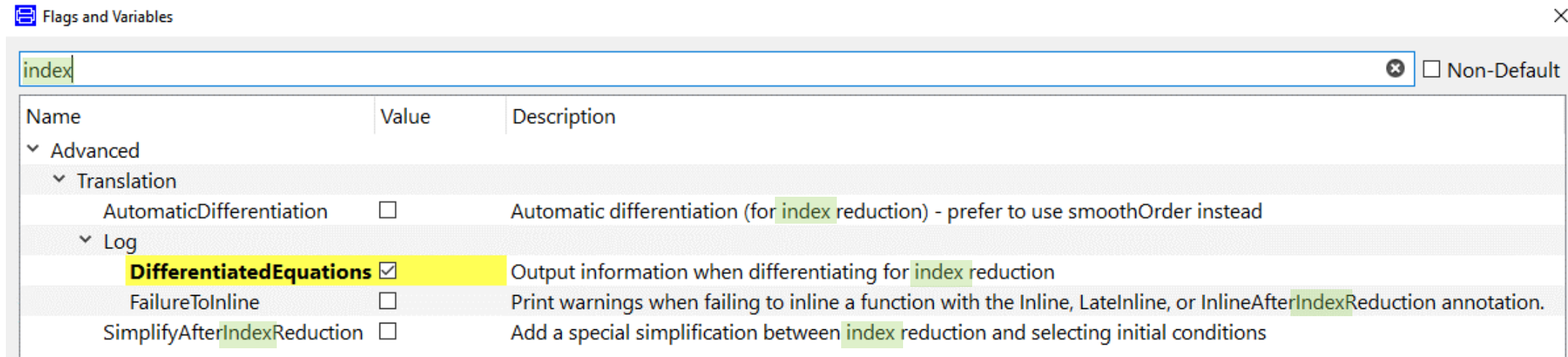
Leichter zugängliche Hilfetexte

- An diversen Stellen in unterschiedlichen Dialogen wurde ein Hilfe-Button eingefügt
- Dies zeigt den Text an, der bislang nur mit Rechtsklick → “What’s This?” erreichbar war



Erweiterte Suche im Flag & Variable Dialog

Die Suche im Flags und Variables Dialog berücksichtigt nun auch die Description:



The screenshot shows the 'Flags and Variables' dialog box with a search bar containing 'index'. The search results are displayed in a table with columns for Name, Value, and Description. The 'DifferentiatedEquations' flag is highlighted in yellow and is checked.

Name	Value	Description
Advanced		
Translation		
AutomaticDifferentiation	<input type="checkbox"/>	Automatic differentiation (for <code>index</code> reduction) - prefer to use <code>smoothOrder</code> instead
Log		
DifferentiatedEquations	<input checked="" type="checkbox"/>	Output information when differentiating for <code>index</code> reduction
FailureToInline	<input type="checkbox"/>	Print warnings when failing to inline a function with the <code>Inline</code> , <code>Lateline</code> , or <code>InlineAfterIndexReduction</code> annotation.
SimplifyAfterIndexReduction	<input type="checkbox"/>	Add a special simplification between <code>index</code> reduction and selecting initial conditions

Bislang wurde nur nach dem Variablenname gesucht

Export FMU Dialog

- Der Dialog zum Exportieren von FMUs wurde übersichtlicher gestaltet.
- Je nach FMU-Typ (ME, CS) kann man dafür verfügbare Solver auswählen.
- Dadurch entfällt die Unklarheit, dass „using Dymola Solvers“ den im Hauptfenster ausgewählten Solver verwendet.

Dymola 2024x

- Type
- Model exchange
 - Co-simulation using Cvode
 - Model exchange, and Co-simulation using Cvode
 - Co-simulation using Dymola solvers

The image displays four screenshots of the 'Export FMU' dialog box, illustrating the changes in the user interface for selecting the FMU type and algorithm.

Screenshot 1 (Left): Shows the 'Type' section with three radio buttons: 'Model exchange' (selected), 'Co-simulation', and 'Model exchange and Co-simulation'. The 'Algorithm' section has a dropdown menu with 'No inline method' selected. Below it, there are radio buttons for '1.0' and '2.0'.

Screenshot 2 (Middle-Left): Shows the 'Type' section with three radio buttons: 'Model exchange', 'Co-simulation' (selected), and 'Model exchange and Co-simulation'. The 'Algorithm' section has a dropdown menu with 'Cvode' selected. Below it, there is a text field 'Current Dymola solver: Sdirk34hw' and a radio button for 'Current inline method: Select in Realtime tab'.

Screenshot 3 (Middle-Right): Shows the 'Type' section with three radio buttons: 'Model exchange', 'Co-simulation', and 'Model exchange and Co-simulation' (selected). The 'Algorithm' section has a dropdown menu with 'Cvode' selected. Below it, there is a text field 'Dymola solver not available with source code export' and a radio button for 'Current inline method: Select in Realtime tab'. The 'Options' section is visible, containing several checkboxes: 'Include source code' (checked), 'Export with string parameters' (unchecked), 'Declare tunable as fixed' (unchecked), 'Copy resources to FMU' (checked), 'Show dialog when exporting' (checked), 'FMU needs no license key' (checked), 'Evaluate false on exported parameters' (unchecked), and 'Cross compile for Linux' (unchecked). The 'Store result' section is also visible.

Screenshot 4 (Right): Shows the 'Type' section with three radio buttons: 'Model exchange', 'Co-simulation', and 'Model exchange and Co-simulation' (selected). The 'Algorithm' section has a dropdown menu with 'Cvode' selected. Below it, there is a text field 'Dymola solver not available with model exchange' and a radio button for 'Current inline method: Select in Realtime tab'.

Beibehaltung der Formatierung

- Weniger Autoformatierungen des Modelica Codes
 - Wenn man ein Modell nach einer Änderung wieder speichert, wird die Textformatierung (z.B. Einrückungen etc.) besser beibehalten.
 - Dies ist für eine bessere Vergleichbarkeit von Modellständen z.B. in Versionierungstools relevant, die sonst z.B. Whitespace\Zeilenumbruchs Änderungen feststellen würden.
 - Mit der neuen BuiltIn-Funktion *"updateModelicaFormatting"* kann der Quellcode formatiert werden. z.B. *updateModelicaFormatting("Modelica.Blocks.Examples.PID_Controller")*

```
function updateModelicaFormatting "update Modelica text formatting to follow the standard by removing formatting for"  
  input String className;  
  input Boolean topClass = true "Ensure that entire contents of mo-file is not indented.";  
  input Boolean publicProtected = true "The keywords public/protected should not be indented.";  
  input Boolean declarations = true "Declarations should be indented two spaces.";  
  input Boolean trailingSpace = true "Ensure that there are no trailing spaces.";  
  output Boolean ok;
```

- equations und annotations werden bei der Formatierung nicht berücksichtigt.

Beibehaltung der Formatierung

Dymola 2024x

```

PID_Controller x +
Modelica.Mechanics.Rotational.Sources.ConstantTorque loadTorque (
  tau_constant=10, useSupport=false) a;
Modelica.Blocks.Continuous.LimPID PI (
  k=100,
  // Ti=0.1,
  yMax=12,
  Ni = 0.1, initType =Modelica.Blocks.Types.Init.SteadyState,
  controllerType=Modelica.Blocks.Types.SimpleController.PI
  ,Ti = 0.1

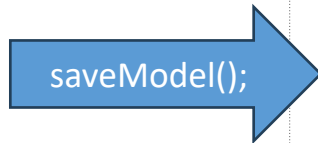
  , limiter(
    u(start
    = 0)
  ),
  Td=0.1) annotation (

  Placement(
    transformation(extent={{-56,
    -20},{-36,0}})
  )
);

initial equation
der
( spring.w_rel
) = 0
;

equation

```



```

PID_Controller x +
Modelica.Mechanics.Rotational.Sources.ConstantTorque loadTorque (
  tau_constant=10, useSupport=false) a;
Modelica.Blocks.Continuous.LimPID PI (
  k=100, yMax=12,
  Ni = 0.1, initType =Modelica.Blocks.Types.Init.SteadyState,
  controllerType=Modelica.Blocks.Types.SimpleController.PI,
  Ti = 0.1,
  limiter(
    u(start
    = 0), Td=0.1) annotation (
  Placement(
    transformation(extent={{-56,
    -20},{-36,0}})));
  // Ti=0.1,

initial equation
der
( spring.w_rel
) = 0;

equation

```

Beibehaltung der Formatierung

Dymola 2024x Refresh 1



```

PID_Controller x +
Modelica.Mechanics.Rotational.Sources.ConstantTorque loadTorque (
  tau_constant=10, useSupport=false) ;
Modelica.Blocks.Continuous.LimPID PI (
  k=100,
  // Ti=0.1,
  yMax=12,
  Ni = 0.1, initType =Modelica.Blocks.Types.Init.SteadyState,
  controllerType=Modelica.Blocks.Types.SimpleController.PI
  ,Ti = 0.1

  ,
  limiter(
    u(start
      = 0)
  ),
  Td=0.1) annotation (

  Placement(
    transformation(extent={{-56,
      -20},{-36,0}})
  )
);

initial equation
der
( spring.w_rel
  ) = 0
;

equation

```

```

PID_Controller x +
Modelica.Mechanics.Rotational.Sources.ConstantTorque loadTorque (
  tau_constant=10, useSupport=false) ;
Modelica.Blocks.Continuous.LimPID PI (
  k=100,
  // Ti=0.1,
  yMax=12,
  Ni = 0.1, initType =Modelica.Blocks.Types.Init.SteadyState,
  controllerType=Modelica.Blocks.Types.SimpleController.PI,
  Ti = 0.1,

  limiter(
    u(start
      = 0)),
  Td=0.1) annotation (

  Placement(
    transformation(extent={{-56,
      -20},{-36,0}}));
initial equation
der
( spring.w_rel)= 0;

equation

```

BETA: Abhängigkeiten in Dialogen

Neuer Beta-Flag:

Name	Value	Description
<ul style="list-style-type: none"> ▼ Advanced <ul style="list-style-type: none"> ▼ Beta 		
EnableUsingModifierLogic	1	0 - always old variant, 1 smart, 2 always new modifier variant; to get enable depending on class parameters to work

Beispiel:

```
model ExampleComponent
  parameter Real x = 1
    annotation (Dialog( enable = enableIntComponent > 0 ));
  parameter Integer enableIntComponent = -1;
end ExampleComponent;
```

Parameters

x	1
enableIntComponent	-1

Ob der Wert von x angepasst werden darf (Dialog(enable=...)), wird bei einer Änderung von *enableIntComponent* nun unmittelbar angepasst.

Bislang musste das Dialogfenster geschlossen und neu geöffnet werden, damit die Änderung in Kraft tritt.

Das funktioniert zum Beispiel auch, wenn die Bedingung von einem replaceableModel abhängt und dieses neu deklariert (redeclare) wird.

Skripting & Versionierung

Python Interface

- Das Dymola-Python Interface wird nun als *.whl File statt *.egg File ausgeliefert
- Es kann **entweder** wie gewohnt dem **python Suchpfad** hinzugefügt werden:

```
C:\work_MS\DymolaVersions\ReleaseNotes2024xRF1>python
>>> import os, sys
>>> sys.path.insert(0,r"C:\Program Files\Dymola 2024x Refresh 1\Modelica\Library\python_interface\dymola-2024.1-py3-none-any.whl")
>>> from dymola.dymola_interface import DymolaInterface
>>> _
```

- **oder** via **pip** installiert werden:
 - 1. (einmalig)

```
C:\work_MS\DymolaVersions\ReleaseNotes2024xRF1>python -m pip install "C:\Program Files\Dymola 2024x Refresh 1\Modelica\Library\python_interface\dymola-2024.1-py3-none-any.whl"
Processing c:\program files\dymola 2024x refresh 1\modelica\library\python_interface\dymola-2024.1-py3-none-any.whl
Installing collected packages: dymola
Successfully installed dymola-2024.1
```

- 2. (bei jeder Nutzung des Interfaces):

```
C:\work_MS\DymolaVersions\ReleaseNotes2024xRF1>python
>>> from dymola.dymola_interface import DymolaInterface
>>>
```

Python Interface

- Python 2.x Support wurde beendet, dafür werden nun auch Python Versionen ≥ 3.7 unterstützt
- Das Python-Interface enthält nun die Funktion *SetVariable*, um den Wert eines Parameters in einem übersetzten Modell zu ändern und die Funktion *updateModelicaFormatting*, um den Modelica Source Code im Text-Layer zu formatieren
- Die Funktion *list* wurde in *listvariables* umbenannt, um name-clashes mit der python-builtin-Funktion *list* zu vermeiden. Sie gibt alle Flags aus.
- Das Argument *debug* beim Starten des Dymola-Python Interfaces entfällt. Es war zum debuggen den Interfaces gedacht.

Scilab Dymola Interface

- Scilab ist eine OpenSource Software nach dem Vorbild von Matlab, verfügbar unter <https://www.scilab.org/>
- Scilab wurde von Dassault erworben und wird OpenSource weiterentwickelt
- Mit Dymola 2024x Refresh1 werden scilab Funktionen mitgeliefert, um Dymola .mat-Files in Scilab einzuladen.
(C:\Program Files\Dymola 2024x Refresh 1\scilab)
- In Scilab kann dann z.B. ein Postprocessing durchgeführt werden
- Die Nutzung ist jedoch noch sehr fehleranfällig, insbesondere dymbrowse()



```
Scilab 2024 0.0 Console
--> exec("C:\Program Files\Dymola 2024x Refresh 1\scilab\dymtools\loader.sce", -1);
Loading Dymola simulation results post-processing tools
Load help
Run 'dymbrowse()' to launch GUI.
--> UnnamedA = dymload("C:\work_MS\Dymola_Work\UnnamedA.mat");
"> C:\work_MS\Dymola_Work\UnnamedA.mat loaded."
--> x=dymget(UnnamedA, "x");
--> t=dymget(UnnamedA, "Time");
--> plot(t, x);
-->
```

Interface laden

Mat-File laden

Trajektorien auslesen

plotten

BuiltIn Funktion SetDymolaCompiler() verbessert

Die builtIn Funktion SetDymolaCompiler wurde verbessert:

1. Sie gibt eine Warnung, wenn ein unbekannter Compiler angegeben wurde:

```
SetDymolaCompiler("mingw")
Warning: The compiler mingw is unknown.
```

2. Mit dem neuen Argument *mergeSettings* = true (=default) werden nur die im Befehl gemachten Angaben überschrieben und alle anderen beibehalten:

Bsp: Beim Umschalten von VisualStudio auf WSL geht der VS-Pfad nicht verloren

Dymola 2024x Refresh 1

```
SetDymolaCompiler("vs", {"MSVCDir=C:\Program Files\....."})
GetDymolaCompiler()
= "vs", {"MSVCDir=C:\\Program Files\\....."}
SetDymolaCompiler("wsl", {"WSLPath=C:/Windows/System32/wsl.exe"})
GetDymolaCompiler()
= "wsl", {"MSVCDir=C:\\Program Files\\.....",
"WSLPath=C:/Windows/System32/wsl.exe"}
SetDymolaCompiler("vs")
```

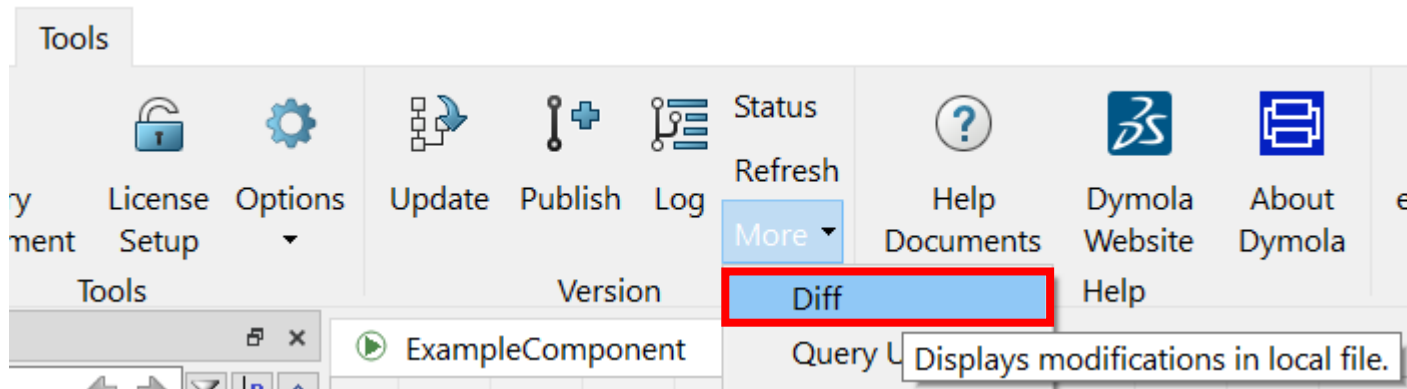
Dymola 2024x

```
SetDymolaCompiler("vs", {"MSVCDir=C:\Program Files\....."})
GetDymolaCompiler()
= "vs", {"MSVCDir=C:\\Program Files\\....."}
SetDymolaCompiler("wsl", {"WSLPath=C:/Windows/System32/wsl.exe"})
GetDymolaCompiler()
= "wsl", {"WSLPath=C:/Windows/System32/wsl.exe"}

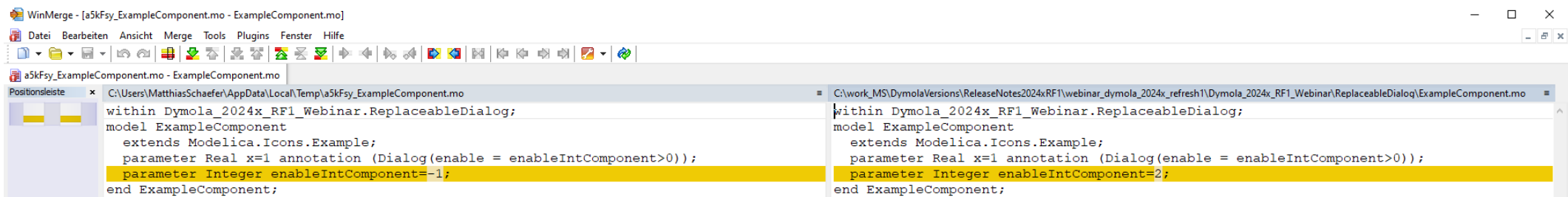
SetDymolaCompiler("vs", {"MSVCDir=C:\Program Files\....."})
```

3. Der Befehl ist nun auch unter Linux verfügbar

Vergleich von Versionierungsständen (git)



Die Anbindung des Versionierungstools git wurde um die git diff Funktionalität erweitert. Zur Darstellung wird ein externes Tool (kdiff3 oder winMerge) benötigt:

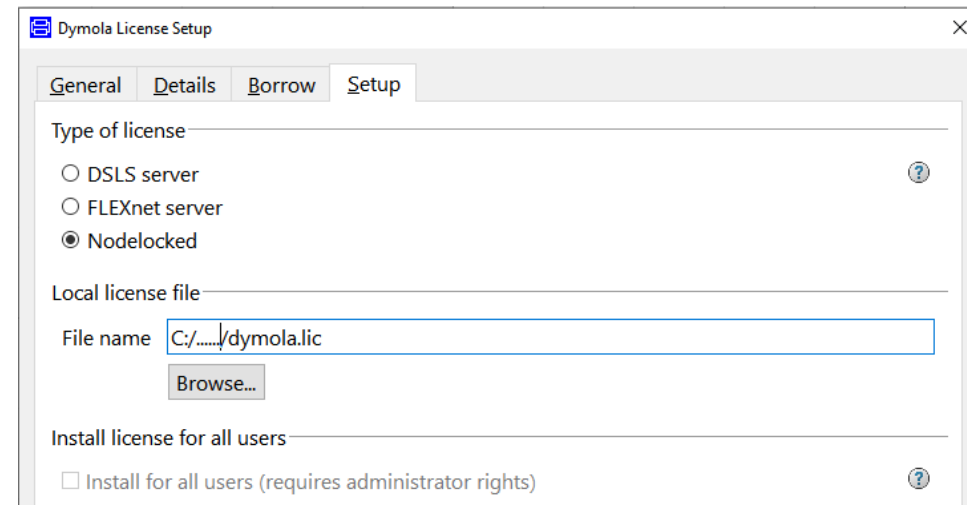


Es wird der auf der Festplatte gespeicherte Modellstand mit dem Stand im Versionierungstool verglichen

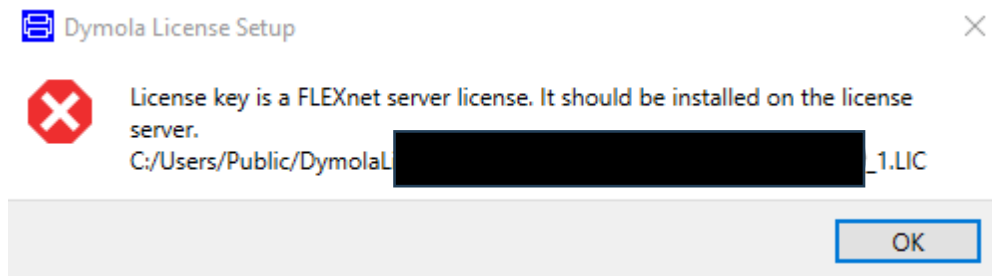
Lizenzierung

Handling von NodeLocked Lizenzen

- Beim Einrichten von NodeLocked Lizenzen muss nicht mehr zwischen FlexNet und DSLS unterschieden werden. Dymola erkennt das nun automatisch:

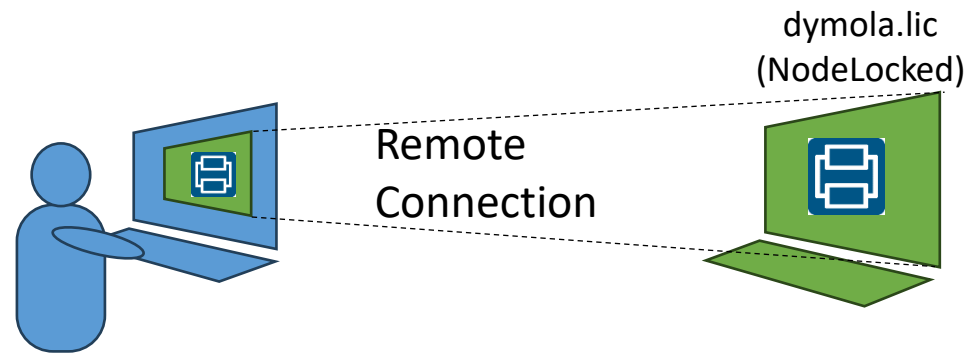


- Dymola erkennt auch, wenn versehentlich versucht wird eine Server-Lizenz als nodelocked einzuspielen:



Remote Desktop Connection mit NodeLocked Lizenzen

- Neu ausgestellte NodeLocked Lizenzfiles ermöglichen nun eine Nutzung von Dymola z.B. via Remote Desktop Connection:



- Bislang war dafür eine Server-Lizenz erforderlich



Wenn Sie dies nutzen wollen, erfragen Sie ein neues Lizenzfile unter sales@ltx.de

Modellbibliotheken

Dymola & Modelica Standard Library

Dymola Version	MSL Version
Dymola 2025x	4.1.0 ?
Dymola 2024x Refresh 1	4.0.0
Dymola 2024x	4.0.0
Dymola 2023x Refresh 1	4.0.0
Dymola 2023x	4.0.0
Dymola 2023	4.0.0
Dymola 2022x	4.0.0
Dymola 2022	4.0.0
Dymola 2021x	4.0.0
Dymola 2021	3.2.3
Dymola 2020x	3.2.3
Dymola 2020	3.2.3
Dymola 2019 FD01	3.2.2
Dymola 2017	3.2.2
Dymola 2016 FD01	3.2.1
Dymola 2014 FD01	3.2.1
Dymola 2014	3.2

MSL Version	Release Date	# Models/Blocks	# Examples	# Functions
4.1.0	2024-06-15 (geplantes Release-Datum)			
4.0.0	2020-06-04	1417	512	1219
3.2.3	2019-01-23	1288	404	1227
3.2.2	2016-04-03	1600	*	1350
3.2.1	2013-08-14	1360	*	1280
3.2	2010-10-25	1280	*	910
3.1	2009-08-14	923	*	615

* # Examples war in # Models/Blocks enthalten

Neuigkeiten MSL 4.0.0:

http://www.ltx.de/download/molib/MSL4.0.0_Neuigkeiten_LTX.pdf

Aktuelle Maintenance Release für MSL 3.2.3:

MSL v3.2.3+build.4 (2020-06-04)

<https://github.com/modelica/ModelicaStandardLibrary/releases>

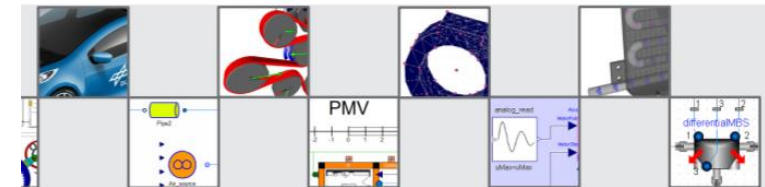
Weitere Libraries (direkt über LTX)

Actuator	Electromechanical Actuators in multiple modeling levels – new
AlternativeVehicles	Hybrid Vehicle Power Trains, Battery-Electric and Fuel-Cell Vehicles
Belts	Static and Dynamic Analysis of Belt Drive Systems
BG_RT	Real-Time Simulation with Standard DAQ Cards
EDrives	Modular, Multi-phase Controlled Electric Drives
EWITAC	Grid-connected Elastic Wind Turbines – new
FlexibleBodies	Flexible Beams, Annular Plates and general FE-based Bodies
HumanComfort	Human Comfort within Air-Conditioned Zones – enhanced
HVAC	Heating and Cooling Systems in Buildings – enhanced
Hydronics	Incompressible Thermo-Hydraulic Fluid Systems
PowerTrain	Vehicle Power Trains and Planetary Gearboxes with Losses
Statistics	Definition of Statistical Variations of Parameters and Variables
TIL	Stationary and Transient Simulation of Thermodynamic Systems – new
Visualization	Advanced, Model Integrated, Offline and Real-Time Visualization



Modelica Libraries Catalog

Version 7



http://www.ltx.de/download/Modelica_Libraries_Catalog_LTX.pdf

[Kurse](#)

[Release Notes](#)

[Download](#)

Nächste Kurstermine

- Online-Kurs an jeweils vier Vormittagen „Einführung zu Dymola und Modelica“
05. - 08. November 2024
- Kurs 2 Tage vor Ort in München „Multidisziplinäre Modellierung und Simulation mit Modelica und Dymola“
Herbst 2024 (genaues Datum wird noch bekannt gegeben)
- Kurs „DLR Visualization Library und Modelica MultiBody“
[Auf Anfrage](#)
- Kurs „Einführung zu OpenModelica“
[Auf Anfrage](#)
- Spezialkurse auf Anfrage, z.B.: [Performance-Workshop](#), [Git&Modelica](#), [FMI&FMPy](#)
Kursthemen für fortgeschrittene AnwenderInnen:
<https://www.ltx.de/download/Kurs/Kursthemen-Fortgeschritten.html>

Aktuelle Termine unter www.kurse.ltx.de

Anfragen und Anmeldungen gerne unter: kurse@ltx.de

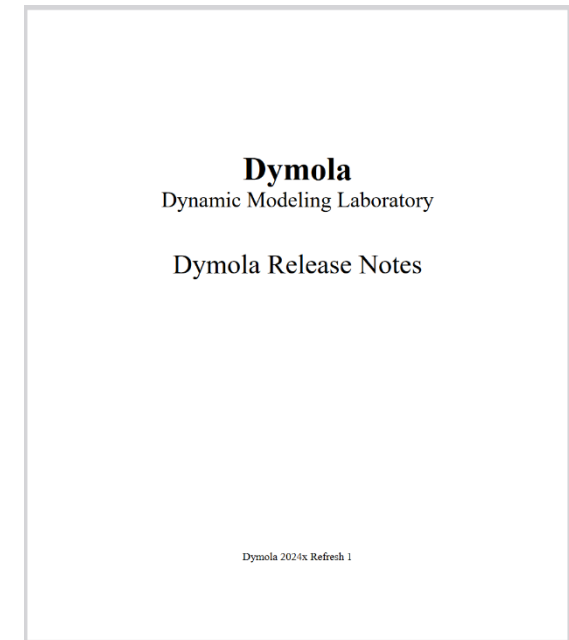
Release Notes

Online:

https://ltx.de/download/Dymola/ReleaseNotes_Dymola2024x_Refresh1.pdf

Dymola:

Help → Documentation → Dymola release notes



Fünfteiliges Handbuch als ein PDF (für Suche):


http://www.ltx.de/download/Dymola/DymolaFullUserManual_Dymola2024x_Refresh1.pdf

Download für Wartungskunden

www.ltx.de/download/Dymola/aktuell

www.ltx.de/download/Dymola/Dymola2024x_Refresh1.html

Noch keine Login-Daten?
Schreiben Sie an sales@ltx.de


Dymola 2024x Refresh 1- Download Info

Wartungskunden erhalten neue Dymola-Versionen kostenfrei zur Verfügung gestellt.

Versionsinfo

Der Download enthält Version Dymola 2024x Refresh 1, Build Date 2024-04-XX

[Ankündigung Webinar](#)
[Vollständige Release Notes](#)
[Installationsanleitung im Nutzerhandbuch](#)

a) Download von Dassault Systèmes

Falls Sie bereits über einen "DS Passport" verfügen (d.h. Zugangsdaten aus vorherigen Media Orders), können Sie Updates direkt vom Hersteller herunterladen:
[Software-Portal bei Dassault Systèmes](#).
Sie finden Dymola dort in der Kategorie "CATIA".

b) Download von LTX

Windows

[Download Dymola 2024x Refresh 1 für Windows](#)
Dateigröße: 1.218.982.043 Bytes (1.13 GB)
SHA1 Prüfsumme: EC03A7B573EADCA731E54968FF1EE6C034072D64
MD5 Prüfsumme: 99789DAEB30F5E1E47717772803D7B13

Linux

[Download Dymola 2024x Refresh 1 für Linux](#)
Dateigröße: 1.064.634.007 Bytes (0.99 GB)
SHA1 Prüfsumme: 86534DB6B6C96CE8A65B51AD63230A71C09E3574
MD5 Prüfsumme: EE1F90B00E0B5238A71606167ED559C8

Zugangsdaten

Wartungskunden erhalten von uns einen Username und Passwort per E-Mail. Solange Sie die Wartung nicht kündigen, sind diese Zugangsdaten auch für künftige Dymola-Versionen gültig.

Sollten Sie noch keine Zugangsdaten von uns erhalten haben, wenden Sie sich bitte an sales@ltx.de

Kontakt

sales@ltx.de – Vertriebsanfragen, Wartungsangebote&-Rechnungen, Lizenz-Keys

support@ltx.de – Technischer Support (SR), Verbesserungsvorschläge (ER)

LTX Simulation GmbH

Wohlfahrtstraße 21 b
80939 München
Deutschland

Tel.: +49 89 20 97 00 55

info@ltx.de

www.ltx.de



Right click icon made by Freepik from www.flaticon.com
Double click icon made from <http://www.onlinewebfonts.com/icon> is licensed by CC BY 3.0