Dymola

Dynamic Modeling Laboratory

Installation

Contents: Chapter 6 “Appendix – Installation” extracted from the manual “Dymola User Manual Volume 1”.

March 2015 (Dymola 2016)

The information in this document is subject to change without notice.

Dassault Systèmes AB
Ideon Gateway
Scheelevägen 27 – Floor 9
SE-223 63 Lund
Sweden

E-mail: http://www.3ds.com/support
URL: http://www.Dymola.com
Phone: +46 46 270 67 00

Contents

1 Appendix — Installation 5

1.1 Installation on Windows 6

1.1.1 Dymola as 32-bit and 64-bit application 6

1.1.2 Installing the Dymola software 6

1.1.3 Installing a C compiler 11

1.1.4 Installing the Dymola license file 14

1.1.5 Additional setup 18

1.1.6 Changing the setup of Dymola 27

1.1.7 Removing Dymola 28

1.1.8 Installing updates 28

1.2 Installation on Linux 28

1.2.1 Installing Dymola 29

1.2.2 Additional setup 29

1.2.3 Removing Dymola 30

1.3 Dymola License Server on Windows 31

1.3.1 Background 31

1.3.2 Installing the license server 31

1.3.3 License borrowing 35

1.4 Dymola License Server on Linux 40

1.5 Utility programs 41

1.5.1 Obtaining a host id 41

1.6 System requirements 42

1.6.1 Hardware requirements 42

1.6.2 Hardware recommendations 42

1.6.3 Software requirements 43

1.7 License requirements 44
1.7.1 General 44
1.7.2 License for Dymola – Simulink interface 45
1.7.3 License for real-time simulation 45
1.7.4 Licenses for exporting code 45
1.7.5 Licenses for executing/importing/using code 46
1.7.6 Licenses for libraries in the Dymola library menu 48
1.7.7 Licenses for libraries not in Dymola library menu 49
1.8 Troubleshooting 49
1.8.1 License file 49
1.8.2 Compiler problems 51
1.8.3 Simulink 53
1.8.4 Change of language 54
1.8.5 Other Windows-related problems 54
2 Index 57

1 Appendix — Installation

This chapter describes the installation of Dymola on Windows and Linux, and related topics.

The content is the following:

In section 1.1 **”Installation on Windows”** starting on page 6 the installation on Windows is described, including installation of Dymola software, C compiler and license (sharable or node-locked). The sub-section “Additional setup” starting on page 18 treats specific issues as installing Dymola as administrator on a computer that should be used by non-administrators and remote installation of Dymola. Finally change of setup, removal of Dymola and installing updates are described.

In section 1.2 **“Installation on Linux”** starting on page 28 the installation on Linux is described, in a similar way as the previous section. The sub-section “Additional setup” starting on page 29 describes e.g. compilation of model code and simulation from the command line.

In section 1.3 **“Dymola License Server on Windows”** starting on page 31 the installation of a license server on Windows is described, as is the borrowing of licenses.

In section 1.4 **“Dymola License Server on Linux”** starting on page 40 the installation of a license server on Linux is described, as is the borrowing of licenses.

In section 1.5 **“Utility programs”** starting on page 41 a utility program for finding a host id on a computer is described.

In section 1.6 **“System requirements”** starting on page 42 the hardware and software requirements/recommendations are listed.

In section 1.7 **“License requirements”** starting on page 44 the license requirements for various features are listed.

In section 1.8 **“Troubleshooting”** starting on page 49 the solution to various problems are described. It might be license file problems, compiler problems, issues with Simulink, change of language etc.

1.1 Installation on Windows

This section refers only to the Windows version of Dymola.

To install Dymola the following tasks must be performed:

- Install the Dymola software and libraries.
- Install a C compiler (if it has not been done before).
- Install the Dymola license file.
- Install a license server (sharable license only).

Following installation the user may do additional setup. The installation of updates and removal of Dymola is also described below.

1.1.1 Dymola as 32-bit and 64-bit application

The Dymola program is available both as 32-bit and 64-bit applications. Both are installed when installing Dymola on Windows. The 64-bit Dymola program (and its associated DLLs) is located in the folder `Program Files (x86)\Dymola 2016\bin64` after installation.

1.1.2 Installing the Dymola software

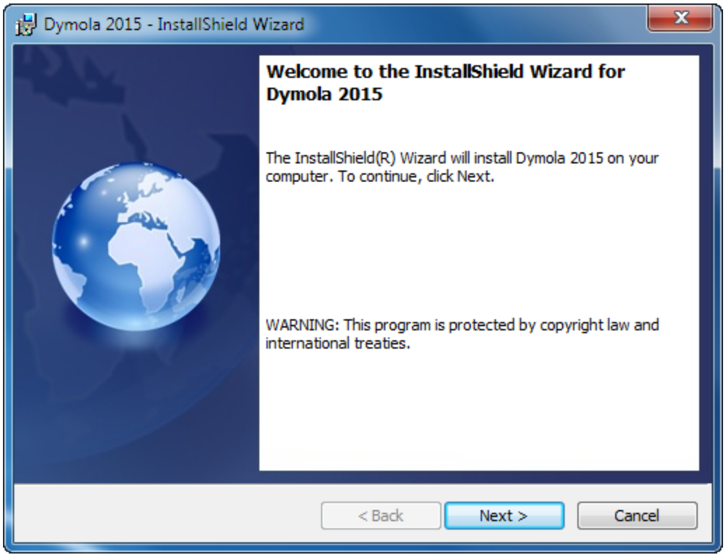
Dymola and appropriate libraries is distributed on a single DVD or downloaded electronically. With electronic download, the DVD-image is provided as two separate .zip files. **Note** that both zip-files must be extracted to the same location before starting the installation.

Starting the installation

Please note that Administrator privileges are required for this installation. When Dymola has been installed, any user can run it.

The installation normally starts when you insert the distribution DVD. If autostart has been disabled, please start `D:\setup.exe` (assuming your DVD drive is labeled D) from Windows Explorer by double clicking on the file or use the **Start** button in Windows, select **Run**, enter `D:\setup.exe` and click **OK**.

Dymola installation setup.



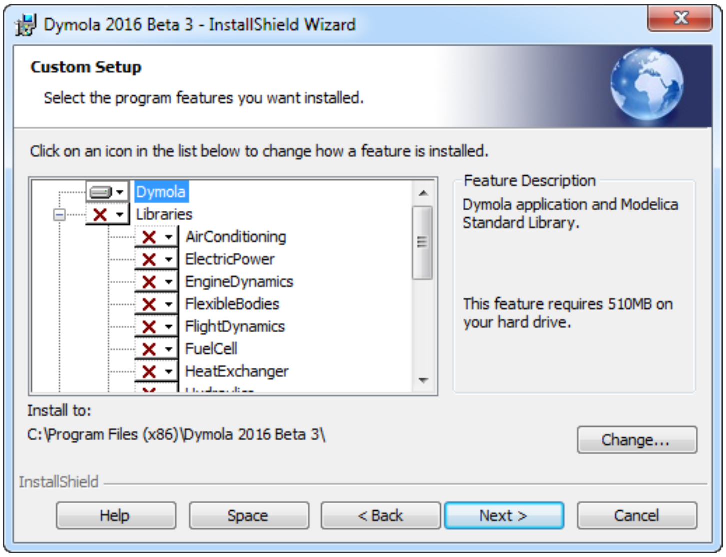
Dymola 2015 - InstallShield Wizard

Welcome to the InstallShield Wizard for Dymola 2015

The InstallShield(R) Wizard will install Dymola 2015 on your computer. To continue, click Next.

WARNING: This program is protected by copyright law and international treaties.

Clicking **Next>** will display license conditions that must be accepted in order to proceed. Accepting by selecting that alternative and then clicking **Next>** will display the following:



Dymola 2016 Beta 3 - InstallShield Wizard

Custom Setup

Select the program features you want installed.

Click on an icon in the list below to change how a feature is installed.

- Dymola
- Libraries
  - AirConditioning
  - ElectricPower
  - EngineDynamics
  - FlexibleBodies
  - FlightDynamics
  - FuelCell
  - HeatExchanger

Feature Description

Dymola application and Modelica Standard Library.

This feature requires 510MB on your hard drive.

Install to:

C:\Program Files (x86)\Dymola 2016 Beta 3\

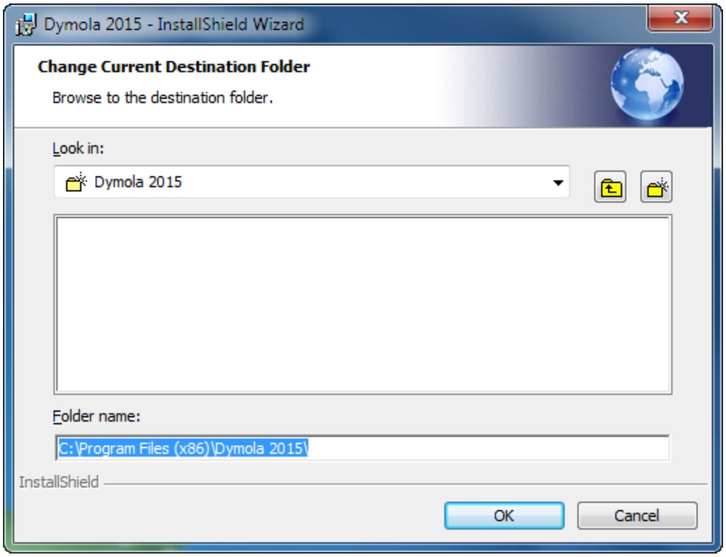
Location of directory

The first choice in the installation procedure is the type of installation and the name of the Dymola installation directory. The default is:

- On 64-bit computers: `C:\Program Files (x86)\Dymola` + the version number of Dymola.
- On 32-bit computers: `C:\Program Files\Dymola` + the version number of Dymola.

This path is displayed under `Install to:`. If the path should be changed, click on the **Change…** button. Here the path can be changed; a change has to be acknowledged by clicking **OK**.

Dymola installation directory.



Dymola 2015 - InstallShield Wizard

Change Current Destination Folder

Browse to the destination folder.

Look in: Dymola 2015

Folder name: C:\Program Files (x86)\Dymola 2015\

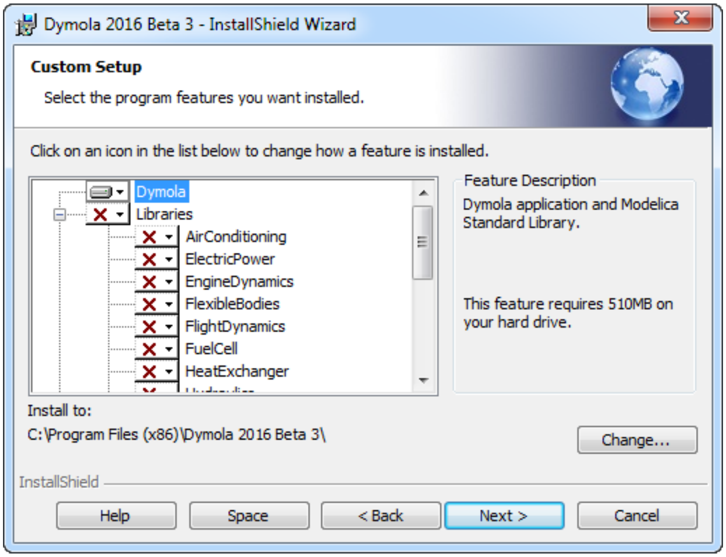
Dymola defines an environment variable DYMOLAWORK which value is the Dymola working directory. This is the default location where Dymola starts.

Dymola by default starts in the directory `My Documents\Dymola` (that subdirectory will be created if it doesn’t exist). Please note that this cannot be an UNC path (i.e. \\server\...).

Selecting components

The second choice is to select optional components of the distribution. By unselecting components some space can be saved.

Component selection.



Dymola 2016 Beta 3 - InstallShield Wizard

Custom Setup

Select the program features you want installed.

Click on an icon in the list below to change how a feature is installed.

- Dymola
- ☒ Libraries
  - ☒ AirConditioning
  - ☒ ElectricPower
  - ☒ EngineDynamics
  - ☒ FlexibleBodies
  - ☒ FlightDynamics
  - ☒ FuelCell
  - ☒ HeatExchanger

Feature Description

Dymola application and Modelica Standard Library.

This feature requires 510MB on your hard drive.

Install to:

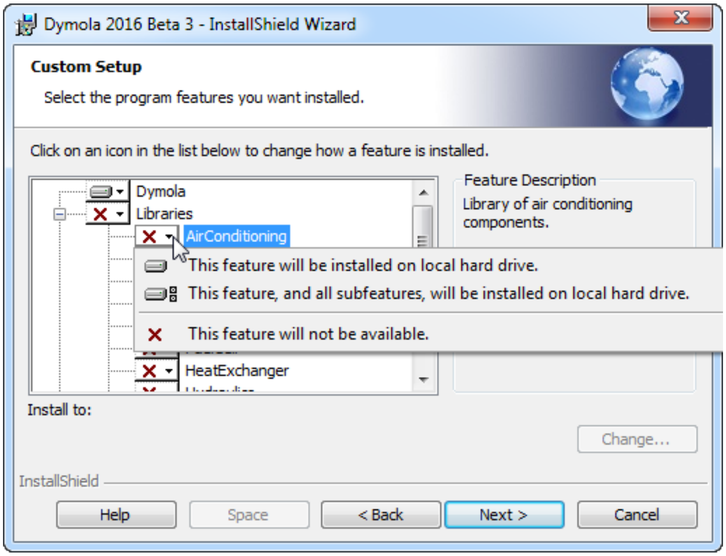
C:\Program Files (x86)\Dymola 2016 Beta 3\

The first alternative **Dymola** is the default contents of the Dymola distribution, including the development environment and the Modelica standard library. This component should always be installed (except when only a license server should be installed).

The **Libraries** section contains several commercial libraries which require a license option to use. Install libraries according to your current options.

The last section, **License server**, makes it possible to install Dymola license server without having to install Dymola. Please note that the **Dymola** component should be unchecked in that case.

To add/remove a component from the installation, click on it and select the appropriate alternative in the menu.



Dymola 2016 Beta 3 - InstallShield Wizard

Custom Setup

Select the program features you want installed.

Click on an icon in the list below to change how a feature is installed.

This feature will be installed on local hard drive.

This feature, and all subfeatures, will be installed on local hard drive.

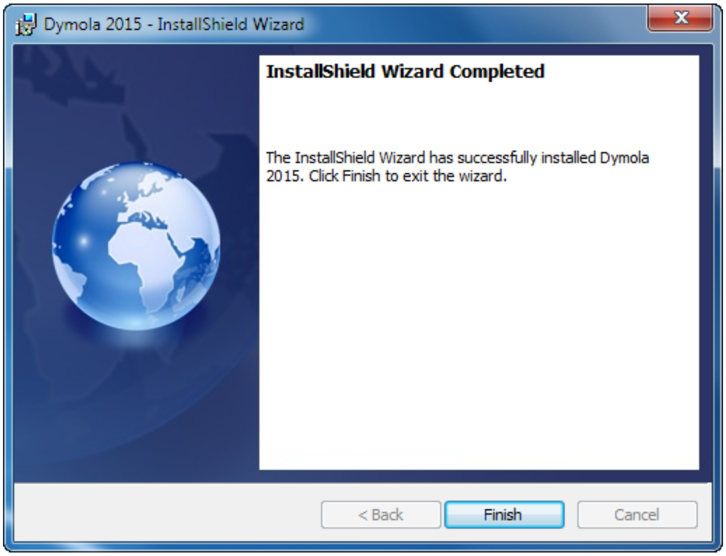
This feature will not be available.

Feature Description

Library of air conditioning components.

When Dymola is successfully installed the following will appear:

Installation of Dymola has finished.



Dymola 2015 - InstallShield Wizard

InstallShield Wizard Completed

The InstallShield Wizard has successfully installed Dymola 2015. Click Finish to exit the wizard.

1.1.3 Installing a C compiler

To translate models in Dymola you must also install a supported C compiler. The C compiler is not distributed with Dymola. The C compiler needs to be installed only once, even if you install multiple versions of Dymola. You can select a Microsoft compiler or a GCC compiler.

Microsoft compilers

Dymola supports Microsoft Visual Studio 2013, both the Professional edition and the Express edition. Dymola also supports older Microsoft compilers (Visual Studio 2012, Visual Studio 2010 and 2008 Professional edition and Express edition, and Visual Studio 2005 Professional edition).

To download the free Visual Studio 2013 Express edition compiler please visit

```
http://www.Dymola.com/compiler
```

where the latest links to Microsoft's website are available. Note that you need administrator rights to install the compiler.

The C compiler can be installed before or after you install the Dymola. You can run Dymola and browse models, but to translate any model you must install the C compiler.

Please note that earlier free versions of the Microsoft compiler are not supported; the reason is that they do not include a full set of Windows libraries. We recommend Visual C++ 2010 or later (see above).

To get a small improvement of the simulation performance, you can activate the global optimization in the compiler, by setting the flag

```
Advanced.Define.GlobalOptimizations = 2;
```

before generating code. (The default value of the flag is 0.)

This flag works the same for all Visual Studio compilers. Note that although the simulation performance is somewhat improved setting this flag, the compilation of the code might take substantially longer time for large models. The setting corresponds to the compiler command `/Og`.

GCC compiler

Dymola 2016 has limited support for the MinGW GCC compilers, with GCC versions compatible with the following:

- 32-bit: MinGW GCC 4.8.1.
- 64-bit: MinGW GCC 4.9.2.

To download any of these free compilers, please visit

```
http://www.Dymola.com/compiler
```

where the latest links to downloading the compilers are available. Needed add-ons etc are also specified here. Note that you need administrator rights to install the compiler.

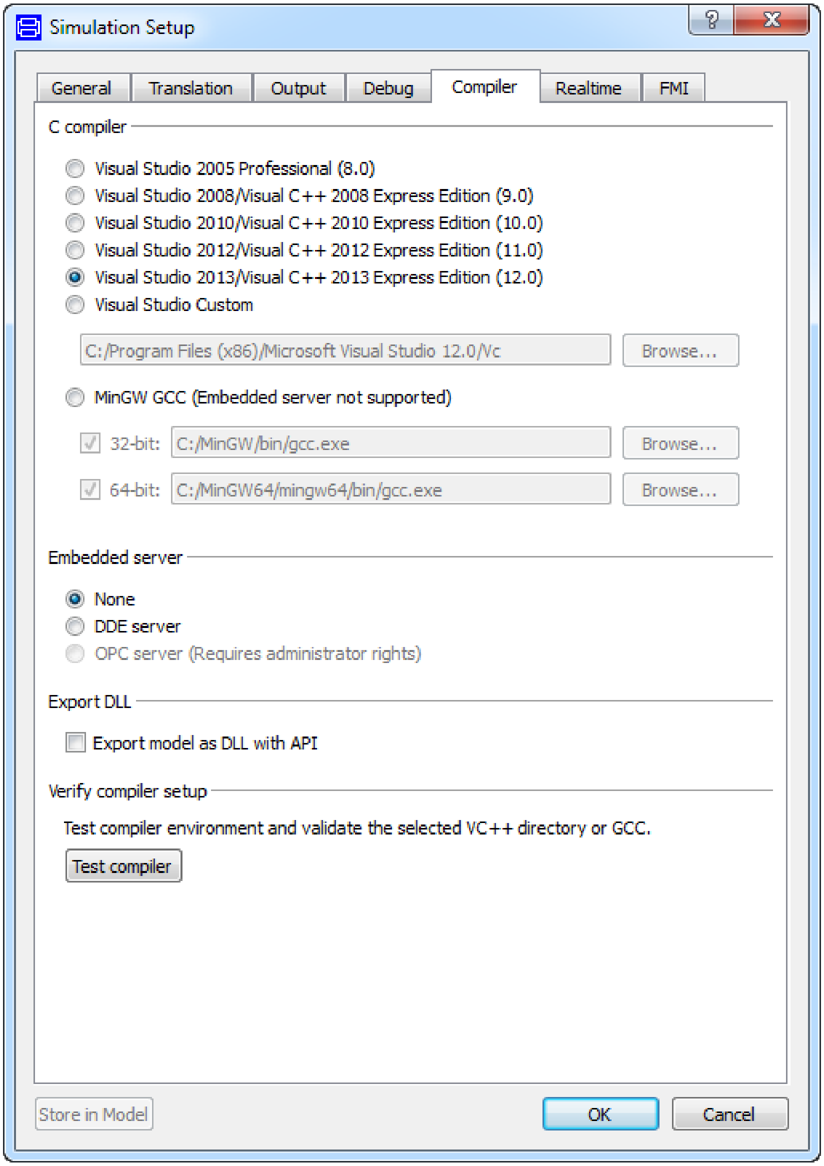
Please note:

- To be able to use other solvers than Lsodar and Dassl, you must also add support for C++ when installing the GCC compiler. Usually you can select this as an add-on when installing GCC.
- There are currently some limitations with GCC:
    - Embedded servers (DDE or OPC servers) are not supported.
    - Commercial libraries: only limited testing has been done.
    - Support for external library resources is implemented, but requires that the resources support GCC, which is often not the case.
    - No support for run-time license.
    - For 32-bit simulation, parallelization is only supported with any of the Lsodar, Dassl, Euler, Rkfix2, Rkfix3, or Rkfix4 algorithms.

Selecting a compiler

Selecting compiler is required.

To change the compiler Dymola uses to translate the model, use the command **Simulation > Setup...** and the **Compiler** tab, see also chapter “Simulating a model”, section “Editor command reference – Simulation mode”, sub-section “Main window: Simulation menu”, command “Simulation > Setup…”. (Below is an example of the **Compiler** tab).



C compiler

○ Visual Studio 2005 Professional (8.0)
○ Visual Studio 2008/Visual C++ 2008 Express Edition (9.0)
○ Visual Studio 2010/Visual C++ 2010 Express Edition (10.0)
○ Visual Studio 2012/Visual C++ 2012 Express Edition (11.0)
● Visual Studio 2013/Visual C++ 2013 Express Edition (12.0)
○ Visual Studio Custom

C:/Program Files (x86)/Microsoft Visual Studio 12.0/Vc

○ MinGW GCC (Embedded server not supported)

☑ 32-bit: C:/MinGW/bin/gcc.exe

☑ 64-bit: C:/MinGW64/mingw64/bin/gcc.exe

Embedded server

● None
○ DDE server
○ OPC server (Requires administrator rights)

Export DLL

□ Export model as DLL with API

Verify compiler setup

Test compiler environment and validate the selected VC++ directory or GCC.

The selected compiler is stored as a per user setting and for the future kept for new installations of Dymola. Switching compiler does not modify Dymola/bin.

Note the importance of the **Test compiler** button. As an example, you cannot see from the menu if you have a valid MinGW compiler available; you must use the **Test compiler** button to see if this is the case.

Classes which contain “Library” annotations to link with external libraries in C are supported for Microsoft Visual Studio compilers. If you link with your own C-libraries you have to recompile them as multi-threaded. The reason is that single-threaded compilation is phased out in Visual Studio 2005, and multi- and single-threaded libraries reading from files are not link-compatible. Thus Dymola only supports linking with multi-threaded libraries in Microsoft Visual Studio compilers. For GCC compilers, see the limitations above.

For information about possible compiler problems, please see the troubleshooting section “Compiler problems” on page 51.

1.1.4 Installing the Dymola license file

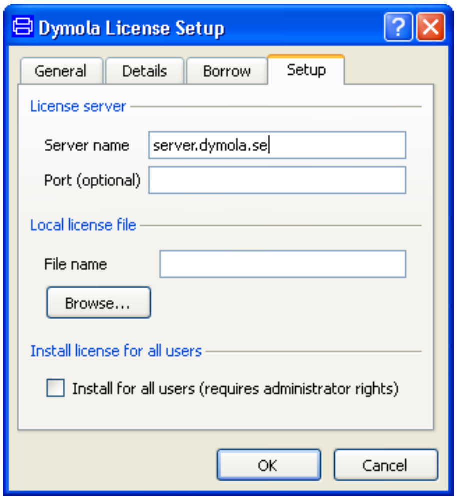
After installation Dymola will initially start in “demo” mode. While running in demo mode you can continue with installing the license file.

Setting up a sharable license

Sharable licenses are requested by Dymola from a license server. The information normally required on the client computer is just the name (or IP number) of the license server.

Start Dymola and select **Help > License…**, and then the **Setup** tab. Enter the name or IP number of the server. If so instructed by the system administrator, also enter the port number. By default leave this field empty.

License server setup.



Server name: server.dymola.se

Port (optional):

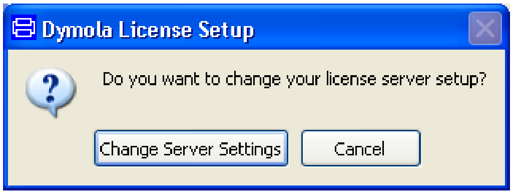
File name:

□ Install for all users (requires administrator rights)

If you want to use redundant servers, you can add three server names/IP numbers, separated by space. Note that one or three servers must be specified.

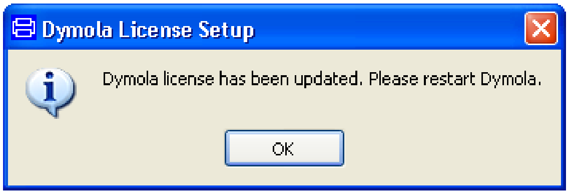
You have the option of installing the license file only for the currently logged in user, or for all users on this computer. The latter requires administrator rights.

Click on the **OK** button. Dymola will ask for confirmation before overwriting your old license information.



Do you want to change your license server setup?

After changing the license server setup you must restart Dymola to use the new server.



Dymola license has been updated. Please restart Dymola.

Installing a node-locked license

Node-locked licenses are stored locally on the computer running Dymola and are not shared with other computers.

Obtaining a host id

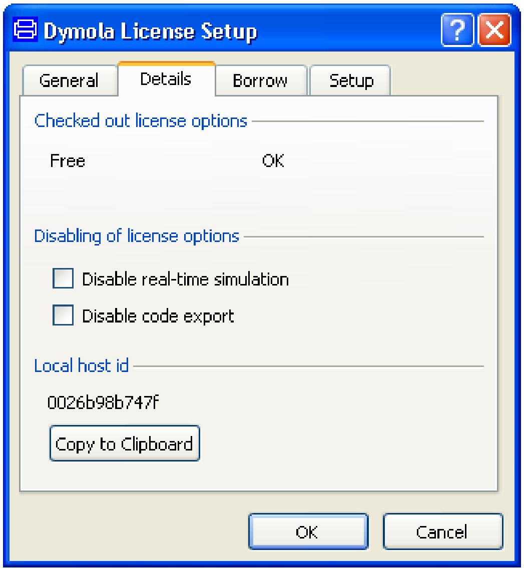
To order a *node-locked license key*, the relevant host id of the computer where Dymola should run must be supplied to your Dymola distributor. The license that you will receive will contain this information.

There are two ways finding out this host id, depending on whether a Dymola demo is installed before or not. The host id can always be fond using the utility program `hostid.exe`. Please see section “Obtaining a host id” on page 41 for more information about this program.

If the Dymola demo has already been installed, Dymola can be used to find the host id. Start Dymola and select **Help > License…**, and then the **Details** tab. Click on **Copy to Clipboard** to copy the local host id.

Please note that some laptops present different host id´s depending on whether they are connected to a docking station or not. In such a case, please copy all host id´s.

Local host id of the computer running Dymola.



Dymola License Setup

Checked out license options

Free OK

Disabling of license options

□ Disable real-time simulation

□ Disable code export

Local host id

0026b98b747f

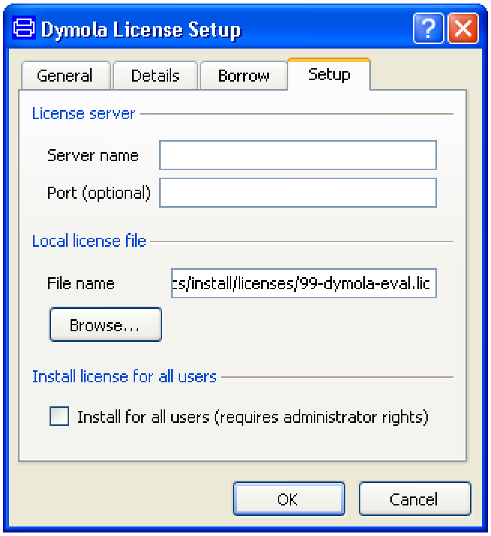
Compose an e-mail containing your local host id (host id´s) and send it to your Dymola distributor.

Installing the node-locked license

When you have received your license file, do save the license somewhere on your computer.

Start Dymola and select **Help > License…**, select the **Setup** tab. Click on the **Browse** button and open the license file you saved. The path of the license file is shown in the dialog.

Specifying the license file.



License server

Server name

Port (optional)

Local license file

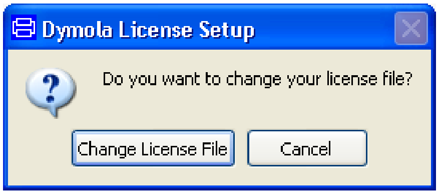
File name :s/install/licenses/99-dymola-eval.lic

Install license for all users

☐ Install for all users (requires administrator rights)

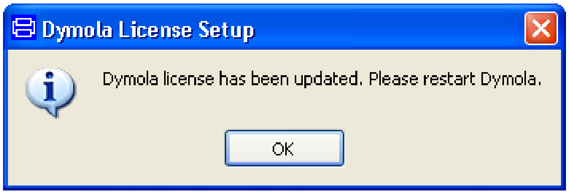
You have the option of installing the license file only for the currently logged in user, or for all users on this computer. The latter requires administrator rights.

Click on the **OK** button. Dymola will ask for confirmation before overwriting your old license information.



Do you want to change your license file?

After changing the license server setup you must restart Dymola to use the new server. You may delete the saved license file, Dymola has created a copy.



Dymola license has been updated. Please restart Dymola.

Run-time licenses

Models developed by users that lack export options can still be run at other computers using a run-time license. Dymola run-time requires the user of the model to have the option `DymolaRuntime`. The license file containing the run-time license should be defined by the environment variable `DYMOLA_RUNTIME_LICENSE`, for example

```
set DYMOLA_RUNTIME_LICENSE=C:\My Documents\dymolaRT.lic
```

For information about license requirements in general, see section “License requirements” starting on page 44.

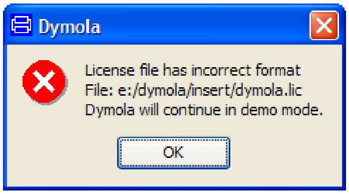
For more specific information about export options in particular, see the manual “Dymola User Manual Volume 1”, chapter 6 “Other Simulation Environments”, section “Code and Model Export”.

Upgrading from Dymola 6.1 and earlier

The license file format of Dymola has been upgraded to include the latest security technology. For that reason, license files for earlier versions of Dymola are not compatible with Dymola 7.0 and later, and license files for Dymola 7.0 and later are not compatible with older versions of Dymola.

If Dymola 7.0 and later finds an old license file at start-up, a diagnostic message about incorrect license file format is displayed. Dymola will then continue execution in demo mode.

Dymola has started with an old license file.



License file has incorrect format
File: e:/dymola/insert/dymola.lic
Dymola will continue in demo mode.

1.1.5 Additional setup

Language

The selection of translation file for Dymola is by default based on the regional language setting.

Please note that currently only English and Japanese are available in Dymola.

If the regional language setting should not be used, there are two ways of overriding it.

The first is to use a command line setting of the language: `-translation <language>`. Two examples are important:

- A customer that wants to run Dymola in Japanese on a machine with regional language setting other than Japanese. This can be done by starting Dymola with the command **“C:\Program Files (x86)\Dymola 2016\bin64\Dymola.exe” –translation ja** (given using a 64-bit Dymola from the default location).
- A customer that wants to run Dymola in English on a machine with regional language setting Japanese. This can be done by starting Dymola with the command **“C:\Program Files (x86)\Dymola 2016\bin64\Dymola.exe” –translation none** (given using a 64-bit Dymola from the default location).

The second way to override the default selection of translation file is to specify what translation file to use, this can be done with the command line option

`–translationfile "<filename.qm>"` when starting Dymola. One specific opportunity here is to use a translation file other that the one in the Dymola distribution. The file can be located anywhere on the machine, since the command line option demands the path of the file to be specified. An example could be to start Dymola with the command **“C:\Program Files (x86)\Dymola 2016\bin64\Dymola.exe” –translationfile”E:\Extra\NewJapaneseTranslationFile.qm”** (given using a 64-bit Dymola from the default location, and a translation file NewJapaneseTranslationFile located in E:\Extra).

Note that command line options can be included in shortcuts to Dymola, see next section.

Creating shortcuts to Dymola

Shortcuts start Dymola in the right directory.

Sometimes it is convenient to create shortcuts to the Dymola program, typically to make Dymola start in the appropriate working directory.

A shortcut is created as follows:

1. Click the right mouse button on the desktop.
2. Select **New > Shortcut** from the popup menu.
3. Browse for the Dymola program (`Program Files (x86)\Dymola 2016\bin\dymola.exe` by default for 32-bit Dymola on a 64-bit computer, `Program Files (x86)\Dymola 2016\bin64\dymola.exe` for 64-bit Dymola on a 64-bit computer).
4. Enter a suitable name and finish the creation of the shortcut.
5. Right-click on the newly created shortcut.
6. Select **Properties** from the popup menu.
7. Select the **Shortcut** tab of the dialog window.
8. Specify a working directory in the **Start in** field.
9. If wanted, add command line options in the **Target** field.
10. Click **OK** to create the shortcut.

Remote installation of Dymola

Dymola (whether downloaded as a zip file or on CD) consists of a number of files (.msi and .cab). Remote installation of dymola.msi is possible using the appropriate tools, such as msiexec. For example, the following command makes a quiet installation of Dymola and all libraries with Modelica version 3:

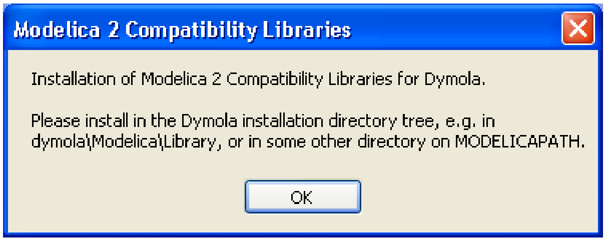
```
msiexec /i dymola.msi INSTALLLEVEL=201 /quiet
```

The value of the `INSTALLLEVEL` property controls which components are installed according to the table:

INSTALLLEVEL	Description
unspecified	Installs Dymola and standard libraries
201	As above and also installs commercial libraries compatible with Modelica language version 3.
301	As above and also installs commercial libraries compatible with Modelica language version 2.2.2
1001	As above and also installs Japanese translations of dialogs and menus

Installing Modelica 2-compatible libraries

Modelica 2-compatible libraries that might be needed to work with an older Modelica model that has not been converted to the new libraries have to be installed separately. However, they are included in the Dymola media as `extra\CompatibilityLibraries.exe`. Running this file will display a message:

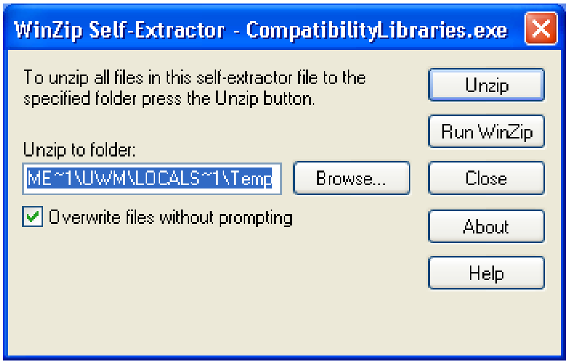


Modelica 2 Compatibility Libraries

Installation of Modelica 2 Compatibility Libraries for Dymola.

Please install in the Dymola installation directory tree, e.g. in dymola\Modelica\Library, or in some other directory on MODELICAPATH.

Selecting **OK** will display:



WinZip Self-Extractor - CompatibilityLibraries.exe

To unzip all files in this self-extractor file to the specified folder press the Unzip button.

Unzip to folder: ME~1\UWM\LOCALS~1\Temp

☑ Overwrite files without prompting

Typically a path to browse to is

```
C:\Program Files (x86)\Dymola 2016\Modelica\Library
```

Selecting **Unzip** will unzip the files to this folder.

Adding libraries and demos to the File menu

Dymola can automatically recognize different libraries in order to e.g. build the **File > Libraries** and **File > Demos** menus. It is very easy to add new libraries and to add new versions of existing libraries.

All information about a library exists in a local file, so it is possible to just "unzip" a subdirectory containing a package, and it will automatically be recognized by Dymola.

No update of a common file is needed, hence no need for special installation scripts. It also makes it easy to delete libraries, just delete the directory.

Dymola will find libraries by searching all directories in the environment variable MODELICAPATH. If not set by the user, MODELICAPATH contains dymola/Modelica/Library.

It is possible to manage the MODELICAPATH from within Dymola, by a built in function

```
AddModelicaPath(path, erase=false);
```

The function appends a directory, specified by the string variable path, to the Modelica path (unless `erase=true`).

An alternative is to use

```
Modelica.Utilities.System.setEnvironmentVariable("MODELICAPATH",
"…");
```

Notes:

- The menus **File > Libraries** and **File > Demos** are rebuilt after a change of MODELICAPATH.
- To keep the change of Modelica path between sessions, this function has to be added in setup.mos. It is currently not possible to store the change of the Modelica path from GUI.

Using library information

Associated with each package is a Modelica script which is automatically located by Dymola at program start. This script can contain a set of commands that describes the package and builds e.g. **File > Libraries**.

The script is called libraryinfo.mos and stored in subdirectory Scripts. Assuming the package is stored in dymola/Modelica/Library/XYZ, the script is called dymola/Modelica/Library/XYZ/Scripts/libraryinfo.mos. Alternatively, the file can be stored in the same directory as the library, e.g. dymola/Modelica/Library/XYZ /libraryinfo.mos.

Note that you must yourself create the scripts for your own libraries if you e.g. want to add them in menus. It is wise to look both below, and at the already present libraryinfo.mos files for libraries already in the **File > Libraries** menu when doing this.

Building menus

There is currently a low-level script command to build libraries and demos menus, e.g.:

```
LibraryInfoMenuCommand(category="libraries",
  text="Hydraulics",
  reference="Hydraulics",
  isModel=true,
  description="Hydraulics Library 4.1 by Modelon",
  version="4.1",
  ModelicaVersion=">=3.2"
  pos=3450);

LibraryInfoMenuCommand(category="demos",
  text="My Demo",
  reference="MyDemo.MyDemoModel",
  isModel=true,
  description="My demo, basic",
  pos=9999);
```

The attributes have the following meaning:

Attribute	**Meaning**
category	Primary menu category (“libraries”, “demos”, or “persistent”)
subCategory[:]	Optional sub-categories
text	Text shown in menu
reference	Model path or command string
isModel	If true, the text is a model path, otherwise a command.
Description	Longer description, for example shown in status bar
version	Version of library (does not apply to demos). See **Important** below.
ModelicaVersion	Required version of Modelica Standard Library, e.g. “> 2”. The value “2” means “>=2”.
Pos	Position in the menu. The menu alternatives are sorted according to this attribute, lowest numbered at the top.

To handle different libraries and groups of libraries, and to make sure Dymola has a consistent ordering of Libraries, Dassault Systèmes allocates ranges of positions to different library vendors. For example, 0 to 999 could be reserved for Dassault Systèmes, 1000 to 1999 for DLR, etc.

Important: For libraries, the version must be specified in the libraryinfo.mos file, and in the corresponding library file (package.mo). The latter is done by the version annotation. The version specified must be the same in both files.

Adding a menu separator

It is possible to add a separator (horizontal line) in the menus. For example,

```
LibraryInfoMenuSeparator(
  category="libraries",
  pos=101);
```

The arguments have the meaning described in the table above.

Loading a package with user-defined menus and toolbars that should not be deleted by File > Clear All

A package with user-defined menus and toolbars where the menus and toolbars should not be deleted by the command **File > Clear All** can be automatically loaded by a libraryinfo.mos file with `category="persistent"` (see also above section).

```
LibraryInfoMenuCommand(
  category="persistent",
  reference="<Class to preload>",
  text="dummy")
```

For more information about creating such menus, see the manual “Dymola User Manual Volume 2”, chapter 7 “User-defined GUI”, section “Extendable user interface – menus, toolbars and favorites”.

License expiration settings

The default behavior of Dymola is:

- To start to warn the user that a license is to expire, 30 days before expiration.
- To continue in demo mode if a license has expired or is faulty.

The behavior can be configured with a command line argument

Consider a user wanting to have the first warning 5 days before the license is expiring, and wanting to terminate Dymola if the license is not found or invalid. Assuming a 32-bit Dymola with default location on MS Widows on a 64-bit computer, Dymola could be started with the following command line using the Command Prompt in Windows:

```
"C:\Program Files (x86)\Dymola 2016\bin\Dymola.exe" /days -5
```

The value (5) controls how many days that should be left to expiration when warning, and the minus before the value is added if Dymola should terminate if the license is not found or invalid.

“-” can be used instead of “/“; the example above will then be:

```
"C:\Program Files (x86)\Dymola 2016\bin\Dymola.exe" -days -5
```

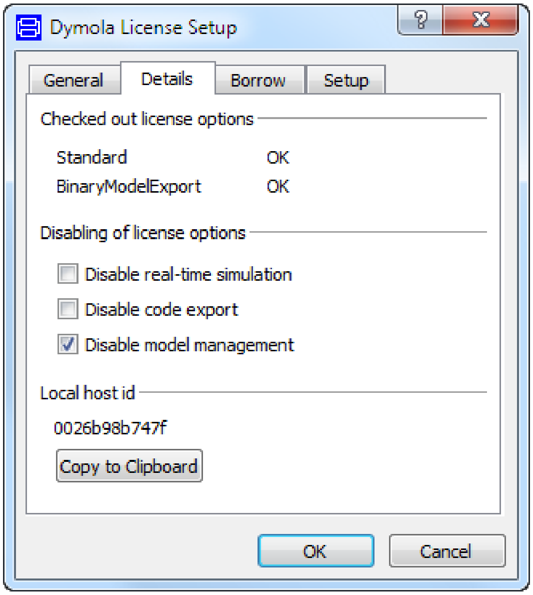
1 day is the minimum start time for warning of license expiration; the warning cannot be completely disabled.

Preventing checking out license options from a license server

It is possible to prevent Dymola from checking out certain license options from the license server, if a sharable license is used. (It is also possible using a node-locked license, e.g. if a user wants to test if a certain model still works without a certain library.)

Using the command **Help > License…** and then looking in the **Details** tab reveals license options currently checked out.

Example of license options checked out.



Dymola License Setup

Checked out license options

Standard OK
BinaryModelExport OK

Disabling of license options

□ Disable real-time simulation
□ Disable code export
☑ Disable model management

Local host id

0026b98b747f

If the user wants to prevent some option from being checked out, it can be done in a number of ways:

- By ticking any of the options in the **Disabling of license options** group in the menu above. The second setting will prevent both BinaryModelExport and SourceCodeGeneration to be checked out. These settings will be remembered between Dymola sessions. They can also be accoplished using the following flags (the below corresponds to prevent check-out of the options, respecively).

  ```
  Advanced.EnableRealtimeSim=false;
  Advanced.EnableCodeExport=false;
  Advanced.EnableModelManagemant=false;
  ```

  For more information concerning real-time simulation and code export, please see the manual “Dymola User Manual Volume 2”, chapter “Other Simulation Environments”.

- By modifying the shortcut to Dymola.
- By starting Dymola with a certatin command line option using the Command Prompt in Windows.

Modifying the shortcut will result in prevention of check out of specified options each time Dymola is started using that shortcut, as. Starting Dymola using a modified command from the command prompt in Windows will only result in prevention of check out of specified options in that session.

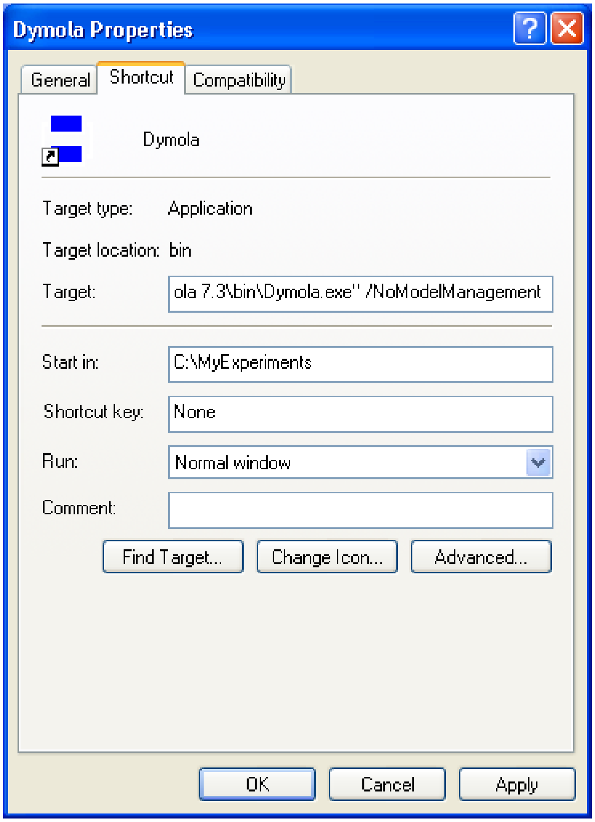
Since the command for prevention of checking out license options is generic, it is very important to use the correct name of the option, including correct use of captitals. The best way is to look at the checked out options using the command above, and mark and copy the name of the option that should not be checked out, to insert that name when using any command.

Modification of the shortcut to Dymola

If a new shortcut is needed, please look at the section “Creating shortcuts to Dymola” above.

To modify the shortcut to prevent checking out a certain option, right-click the shortcut and modify the **Start in**: by adding `<space>/No<optionname>` in the end of the command. If the option ModelManagement in the figure above should not be checked out, the shortcut should be modified like in the figure below.

Modified shortcut.



Dymola Properties

Dymola

Target type: Application

Target location: bin

Target: ola 7.3\bin\Dymola.exe" /NoModelManagement

Start in: C:\MyExperiments

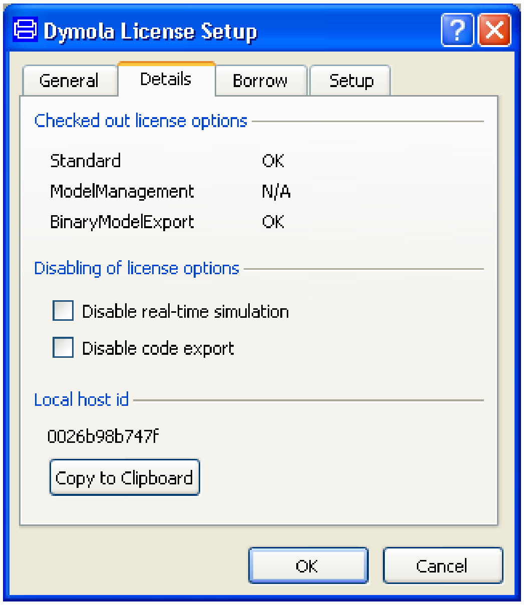
Shortcut key: None

Run: Normal window

Comment:

Closing Dymola and starting it again, the following information will be found in the license tab:

Prevention of checking out a license option.



Checked out license options

Standard	OK
ModelManagement	N/A
BinaryModelExport	OK

Disabling of license options

☐ Disable real-time simulation

☐ Disable code export

Local host id

0026b98b747f

Now ModelManagement will not be possible to check out. As long as the shortcut is not modified, ModelManagement will not be possible to check out from Dymola started by that shortcut.

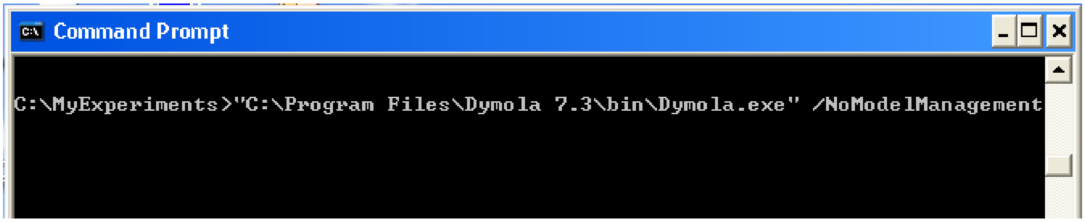
To enable check out of ModelManagement, Dymola must be closed and then restarted using a shortcut without the command line option for ModelManagement.

More than one option can be prevented from check out – just add more strings like the one used. Do not forget the space.

Starting Dymola using a modified command in Command Prompt of Windows

A Windows command prompt can be activated using **Help > All Programs > Accessories > Command Prompt** in Windows.

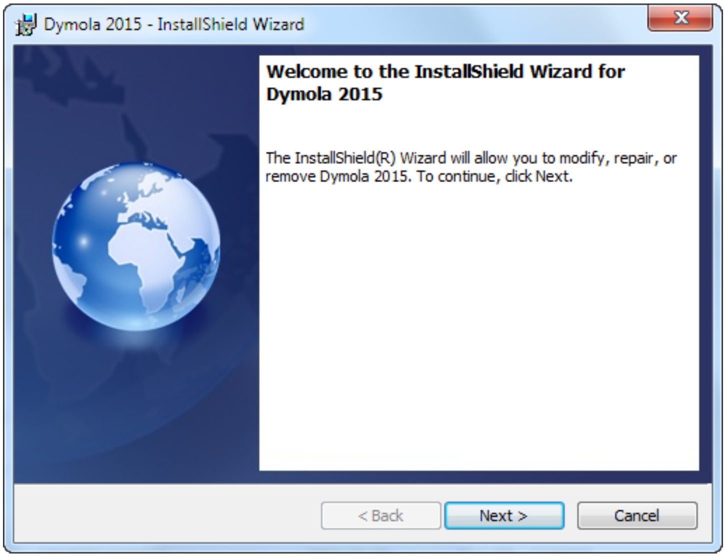
To start one session in Dymola where the license option ModelManagement cannot be checked out like in the example above, the command in the command prompt will look like:



```
C:\MyExperiments>"C:\Program Files\Dymola 7.3\bin\Dymola.exe" /NoModelManagement
```

1.1.6 Changing the setup of Dymola

Under Windows, you can change the setup of Dymola, for example to install additional libraries. Click on the **Start** button in the Taskbar, select **Control Panel** and open **Add or Remove Programs**. Select the relevant version of Dymola and click on the **Change** button.

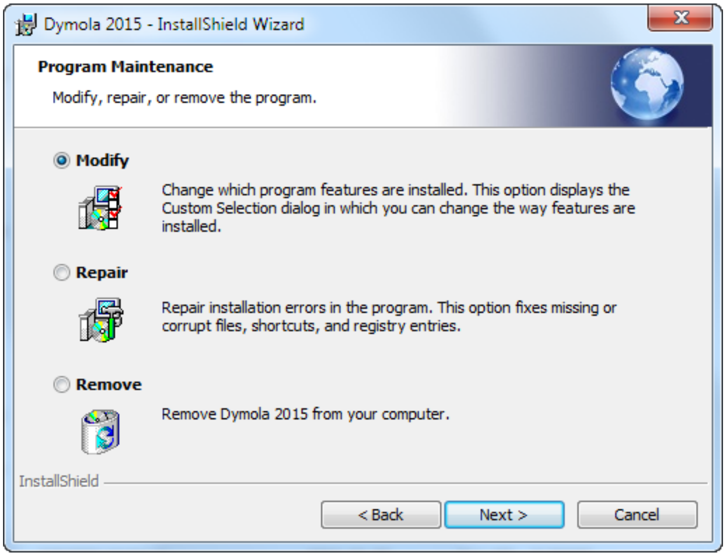


Welcome to the InstallShield Wizard for Dymola 2015

The InstallShield(R) Wizard will allow you to modify, repair, or remove Dymola 2015. To continue, click Next.

Selecting **Next>** will display

Changing Dymola setup.



Program Maintenance

Modify, repair, or remove the program.

● **Modify**

Change which program features are installed. This option displays the Custom Selection dialog in which you can change the way features are installed.

○ **Repair**

Repair installation errors in the program. This option fixes missing or corrupt files, shortcuts, and registry entries.

○ **Remove**

Remove Dymola 2015 from your computer.

To change the setup, choose **Modify**. The rest of the procedure will be the same as when installing Dymola from scratch. Please see previous sections. To restore files in the Dymola distribution that have been deleted by mistake, choose **Repair**. **Remove** will remove the installation.

1.1.7 Removing Dymola

Please see previous section. Do not delete or rename the Dymola directory. Microsoft Windows Installer keeps track of all installed directories and will try to repair if altered. The installation will by default use a directory name that reflects the version of Dymola, but this can of course be changed during setup.

Note that files that you have created in the Dymola distribution directory, for example by running the demo examples, are not deleted when removing Dymola. The remaining files and directories (if any) may be deleted through the Explorer.

1.1.8 Installing updates

Updated versions of Dymola are either distributed on CD, or can be downloaded from a location provided by your sales channel.

Multiple versions of Dymola can be installed, but you cannot install into an existing Dymola directory. Configuration settings and the license file are shared by all installed versions of Dymola.

1.2 Installation on Linux

This section refers only to the Linux version of Dymola.

This section covers Linux-specific parts of the installation. For general items, e.g. how to handle the Dymola installation wizard; please see corresponding section on Windows installation, in particular section “Installing the Dymola license file” on page 14.

The default directory for installation on Linux is `/opt/dymola-<version>-<architecture>`, where the architecture is RPM standard, `i586` for 32-bit, and `x86_64` for 64-bit application. As an example, the default directory for installation of 64-bit Dymola 2016 on Linux is `/opt/dymola-2016-x86_64` (the package manager on the target system however typically allows choosing another default location).

Dymola 2016 runs on SUSE Linux (Release 11), 32-bit and 64-bit, with gcc version 4.3.4, and compatible systems.

In addition to gcc, the model C code can also be compiled by clang. To change compiler, change the variable `CC` in

```
/opt/dymola-<version>-<architecture>/insert/dsbuild.sh
```

(See above concerning `<architecture>`.)

Dymola 2016 is supported as a 32-bit and a 64-bit application on Linux. Corresponding support for 32-bit and 64-bit export and import of FMUs is included.

Notes

- For rendering of jpg files, libjpg62 must be installed.
- 32-bit compilation might require explicit installation of 32-bit libc. E.g. on Ubuntu: `sudo apt-get install g++-multilib libc6-dev-386`

Please also note that you have to use the Optimization library version 2.x or higher to use multi-criteria design optimization on Linux; the older Design.Optimization package does not support multi-criteria design optimization on Linux.

The library UserInteraction is not supported on Linux.

More Linux-specific notes are available using the command

```
man dymola
```

1.2.1 Installing Dymola

Dymola for Linux is distributed as an RPM package. The package is installed using the command

```
# rpm –i name-of-distribution.rpm
```

Optional libraries are installed through separate RPM files.

For installation on e.g. Debian or Kubuntu systems conversion to the `deb` format is required using the `alien` command:

```
# alien –k name-of-distribution.rpm
```

Setup and environment variables

The shell script `/usr/local/bin/dymola-<version>-<architecture>` (see above concerning "-<version>-<architecture>") contains commands to set environment variables before starting Dymola, but will need editing if Dymola is installed in a non-standard location; then the following environment variables must be defined in order to run Dymola:

DYMOLA Directory root of the distribution (`/opt/dymola-<version>-<architecture>`).

DYMOLAPATH Search path for additional Dymola libraries and the license file. The directories of the path may be separated by blanks or colon. `DYMOLAPATH` is optional if the license file is in `$DYMOLA/insert`.

MODELICAPATH Search path for libraries. Concerning the use of MODELICAPATH, please see section "Adding libraries and demos to the File menu" on page 21.

(Dymola defines an environment variable DYMOLAWORK which value is the Dymola working directory.)

1.2.2 Additional setup

Subjects in the corresponding section on Windows are not applicable unless explicitly referenced from here.

Compilation of model code

Dymola produces C code which must be compiled in order to generate a simulation model. On Linux systems we rely on an ANSI/ISO C compiler already installed on the computer.

On Linux systems the compilation of the generated C code is performed by a shell script, `/opt/dymola-<version>-<architecture>/insert/dsbuild.sh` (see above concerning "-<version>-<architecture>"). If necessary this script can be modified to provide special options to the compiler, add application-specific libraries etc. Simulation performance can be improved by tuning the compilations options in this script, however note that the compiler time may increase significantly by doing so.

Dymola supports external C libraries on Linux. Classes which contain "Library" annotations to link with external libraries in C are supported.

Simulation from the command line

The simulator executable `dymosim` can be executed from the shell. To do so the environment variable LD_LIBRARY_PATH must be set:

```
# export LD_LIBRARY_PATH=/opt/dymola-<version>-
<architecture>/bin/lib
```

Security-Enhanced Linux (SELinux) might display the message (below example for 64-bit application):

```
dymosim: error while loading shared libraries: /opt/dymola-
2016-x86_64/bin/lib/libds.so: cannot restore segment prot after
reloc: Permission denied
```

If this message is displayed the following commands must be executed (for this 64-bit example):

```
# chcon –t textrel_shlib_t/opt/dymola-2016-
x86_64/bin/lib/libds.so
# chcon –t textrel_shlib_t/opt/dymola-2016-
x86_64/bin/lib/libGodessMain.so
```

Note that running simulations in the Dymola environment do not require these changes.

Adding libraries and demos to the File menu

Please see corresponding section for Windows installation on page 21.

Preventing checking out options from a license server

In the corresponding section on Windows the alternative of starting Dymola using a modified command is applicable also for Linux (with relevant changes for Linux). Please see page 26.

1.2.3 Removing Dymola

Remove the Dymola distribution by using the `rpm –u` command.

1.3 Dymola License Server on Windows

1.3.1 Background

This section refers only to the Windows version of Dymola.

These are instructions for manually installing a FLEXnet Publisher license server for Dymola on Windows. They only apply to users with a sharable license. For non-sharable licenses (the common case), installation of the license file is automatic.

All files needed to set up and run a Dymola license server on Windows, except the license file, are available in the Dymola distribution, in `Program Files (x86)\Dymola 2016\bin`.

Dymola is installed on all machines which will run the software. On the designated machine, the license server is then installed as described below.

The license server consists of two daemon processes:

- The vendor daemon (called `dynasim.exe`) dispenses licenses for the requested features of Dymola (the ability to run Dymola and various options). This program is specific for Dymola.
- The license daemon (called `lmgrd.exe`) sends requests from application programs to the right vendor daemon on the right machine. The same license daemon can be used by all applications from all vendors, as this daemon processes no requests on its own, but forwards these requests to the right vendor daemon.

If you are already running an application that uses FLEXnet Publisher, you most likely already have a running license daemon. In this case only the vendor daemon (`dynasim.exe`) is required.

Flexera Software recommends that you use the latest version of the FLEXnet Publisher `lmgrd.exe` at all times as it includes bug fixes, enhancements, and assures the greatest level of compatibility with all of your FLEXnet Publisher licensed applications. Flexera Software guarantees that it will work correctly with all earlier versions of FLEXnet Publisher.

Old license daemons cannot be used!

Dymola requires support of FLEXnet Publisher version 11.11. A recent version of `lmgrd.exe` is part of the Dymola distribution.

If needed, the latest available license daemon can be downloaded from the website of Flexera Software when having completed a form:

```
http://mktg.flexerasoftware.com/mk/get/lmgrd_reg
```

1.3.2 Installing the license server

This section describes the simple case where we assume there are no other FLEXnet Publisher license daemons. We also assume that the Dymola program itself should not be installed on the server.

To purchase a license server, the relevant host id of the computer where the license server should run must be supplied to your Dymola distributor before purchasing the license. The license that you will receive will contain this information. To find out the host id of that computer, the utility program `hostid.exe` can be used. Please see section “Obtaining a host id” on page 41 for more information.

1. Before installation of the license server, the Dymola license file (*`filename`*`.lic`) may have to be updated with the actual name (or IP-number) of the server, if the license file contains a line identifying the server:

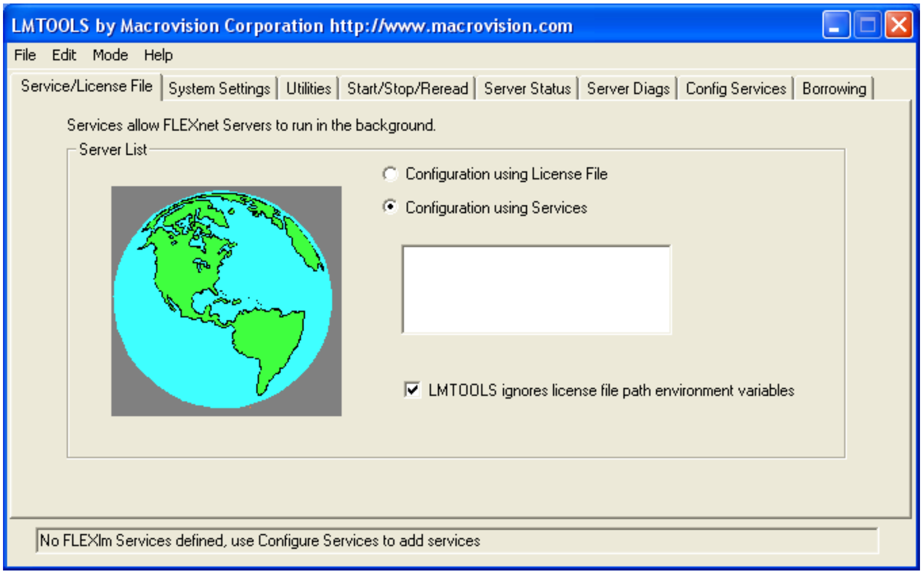
   ```
   SERVER server.name.here 000102DE37CD
   ```

   The part *`server.name.here`* must be changed to the name of the actual server before installing the license file. It should be noted that the last part (the hostid) cannot be edited by the user.
2. Install *only* the Dymola software component **License server** (see beginning of this chapter). A folder will be created containing all needed files, default `C:\Program Files (x86)\Dymola 2016\bin`.
3. Start the utility program `lmtools.exe` (one of the above files).
4. In the **Service/License File** tab:

   a. Select the radio button **Configuration using Services**.

   b. Activate **LMTOOLS ignores license file path environment variables**.

License server setup.



LMTOOLS by Macrovision Corporation http://www.macrovision.com

Services allow FLEXnet Servers to run in the background.

Server List

○ Configuration using License File

● Configuration using Services

☑ LMTOOLS ignores license file path environment variables

No FLEXlm Services defined, use Configure Services to add services

5. In the **Config Services** tab (please see figure on next page):

   a. Enter a new service name, e.g. “Dynasim License Server”.

   b. Enter the path to the license daemon, `Dymola 2016\bin\lmgrd.exe`.

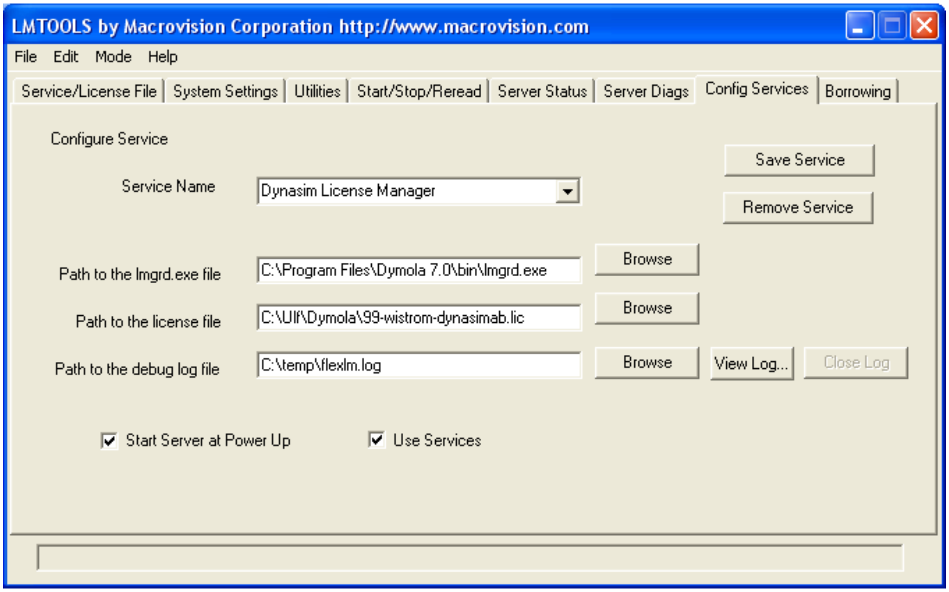
   c. Enter the path to your server license file.

d. Enter the path to a debug log file (anywhere you want).

e. Enable **Use Services** and then **Start Server at Power Up**.

f. Click on **Save Service**. Click on **Yes** to confirm.

Configuration of the license server.



Service Name: Dynasim License Manager

Path to the lmgrd.exe file: C:\Program Files\Dymola 7.0\bin\lmgrd.exe

Path to the license file: C:\Ulf\Dymola\99-wistrom-dynasimab.lic

Path to the debug log file: C:\temp\flexlm.log

☑ Start Server at Power Up ☑ Use Services

6. In the **Start/Stop/Reread** tab:

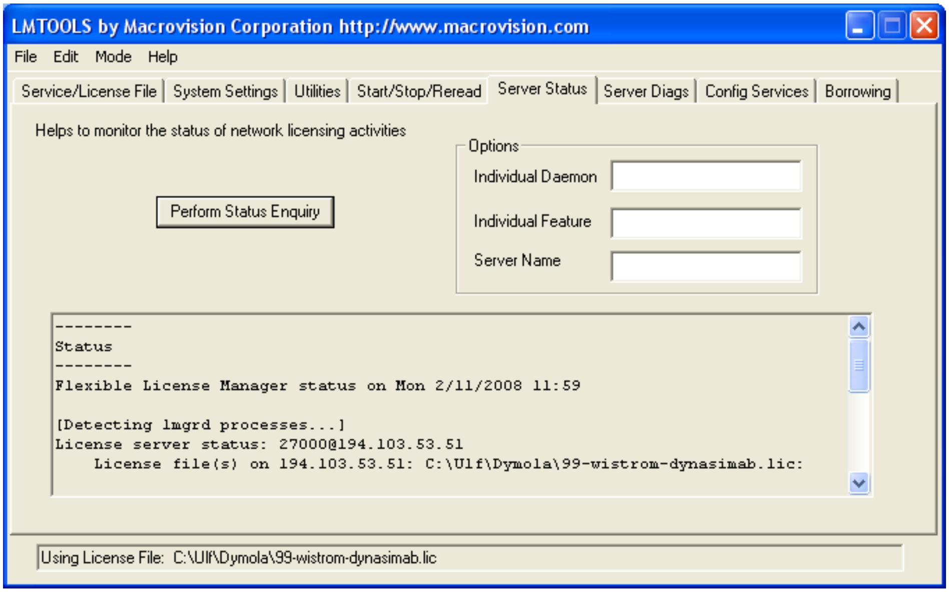
   a. Select the Dynasim license server.

   b. Click on **Start Server**.

7. In the **Server Status** tab:

   a. Click on **Perform Server Enquiry** and check the output of the log window. You should see lines identifying the server processes and what features are available.

Checking the operation of the license server.



```
--------
Status
--------
Flexible License Manager status on Mon 2/11/2008 11:59

[Detecting lmgrd processes...]
License server status: 27000@194.103.53.51
    License file(s) on 194.103.53.51: C:\Ulf\Dymola\99-wistrom-dynasimab.lic:
```

Using License File: C:\Ulf\Dymola\99-wistrom-dynasimab.lic

b. Also check the log file to verify that the server has started and that Dymola features can be checked out. The following is an example of the FLEXnet Publisher logfile:

```
12:30:48 (lmgrd) pid 2728
12:30:48 (lmgrd) Detecting other license server manager (lmgrd) processes...
12:30:48 (lmgrd) Done rereading
12:30:48 (lmgrd) FLEXnet Licensing (v11.4.100.0 build 50818 i86_n3) started
on 194.103.53.51 (IBM PC) (2/11/2008)
12:30:48 (lmgrd) Copyright (c) 1988-2007 Macrovision Europe Ltd. and/or
Macrovision Corporation. All Rights Reserved.
12:30:48 (lmgrd) US Patents 5,390,297 and 5,671,412.
12:30:48 (lmgrd) World Wide Web:  http://www.macrovision.com
12:30:48 (lmgrd) License file(s): C:\Ulf\Dymola\99-wistrom-dynasimab2.lic
12:30:48 (lmgrd) lmgrd tcp-port 27000
12:30:48 (lmgrd) Starting vendor daemons ...
12:30:48 (lmgrd) Started dynasim (pid 4180)
12:30:48 (dynasim) FLEXnet Licensing version v11.4.100.0 build 50818 i86_n3
12:30:48 (dynasim) Server started on 194.103.53.51 for:  DymolaStandard
12:30:48 (dynasim) DymolaAnimation DymolaModelCalibration
DymolaModelManagement
12:30:48 (dynasim) DymolaOptimization DymolaRealtime     DymolaSimulink
12:30:48 (dynasim) DymolaFlexibleBodiesLib DymolaHydraulicsLib
DymolaPowertrainLib
12:30:48 (dynasim) DymolaSmartElectricDrivesLib
12:30:48 (dynasim) EXTERNAL FILTERS are OFF
12:30:48 (lmgrd) dynasim using TCP-port 2606
12:30:56 (dynasim) TCP_NODELAY NOT enabled
10:39:20 (lmgrd) Detecting other lmgrd processes...
10:39:35 (lmgrd) FLEXlm (v7.2c) started on x.x.x.x (3/27/2001)
10:39:35 (lmgrd) FLEXlm Copyright 1988-2000, Globetrotter Software
10:39:35 (lmgrd) US Patents 5,390,297 and 5,671,412.
```

```
10:39:35 (lmgrd) World Wide Web:  http://www.globetrotter.com
10:39:35 (lmgrd) License file(s): C:\DAG\dymola.lic
10:39:35 (lmgrd) lmgrd tcp-port 27000
10:39:35 (lmgrd) Starting vendor daemons ...
10:39:35 (lmgrd) Started dynasim (pid 124)
10:39:36 (dynasim) Server started on x.x.x.x for:DymolaStandard
10:39:36 (dynasim) DymolaSampledLib DymolaLiveObjects DymolaRealtime
10:39:36 (dynasim) DymolaSimulink    DymolaAnimation DymolaSupport
10:39:36 (lmgrd) dynasim using TCP-port 1042
```

The license server should now be correctly configured. Please start Dymola to verify correct operation. The FLEXnet Publisher logfile (see above) should contain additional lines showing what features were checked out. You can also do **Perform Status Enquiry** to check how many licenses are currently checked out.

Note. The license server by default uses the ports 27000-27009. They can be configured if needed, e.g. if there are issues with firewalls.

1.3.3 License borrowing

Overview

Dymola on Windows can support "borrowing", the possibility to transfer a license from a license server to laptop for a limited period of time. If Dymola is used on a computer that is intermittently disconnected from a license server, that license can be issued as a sharable license with borrowing facility. Such a license can be borrowed from a license server via a special checkout and used later to run an application on a computer that is no longer connected to the license server.

For license borrowing, an end user initiates borrowing and specifies the expiration date a borrowed license is to be returned to the sharable license pool. While still connected to the network, the application is run from the client computer. This writes licensing information locally onto the client computer. The client computer can now be disconnected from the network.

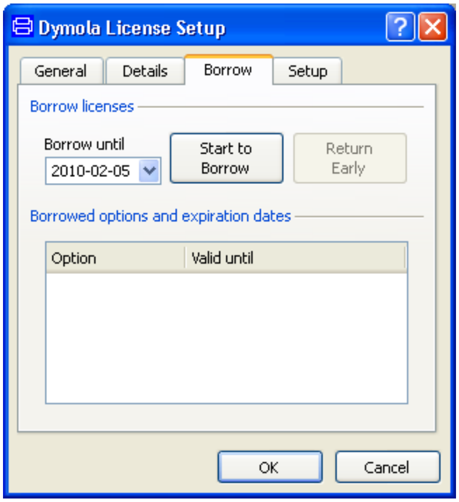
The license server keeps the borrowed license checked out. The client application automatically uses the local secured data file to do checkouts during the borrowing period. Upon the expiration of the borrowing period or the early return of a borrowed license, the local data file no longer authorizes checkouts and the license server returns the borrowed license to the pool of available licenses. No synchronization is required between the license server machine and the client machine when the borrowing period expires.

License borrowing

License borrowing and early returns are performed from Dymola.

In order to borrow, do the following:

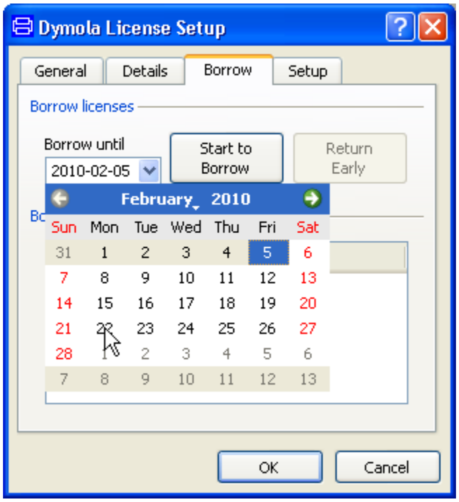
1. While Dymola is connected to the server, use the command **Help > License…**, and select the **Borrow tab**.



Borrow until 2010-02-05

Option	Valid until

2. Select an end date, either by changing the date in the input field for **Last date borrowed** or by clicking on the arrow to display a calendar for selection of date. Clicking the arrow will display:



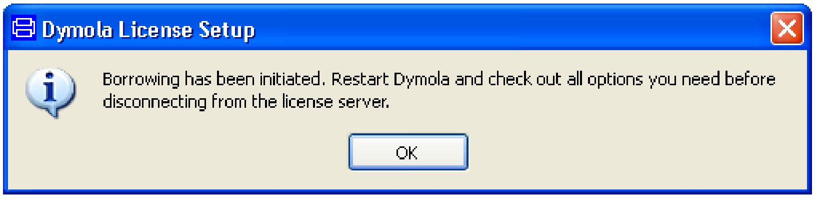
Borrow until 2010-02-05

February 2010

Sun	Mon	Tue	Wed	Thu	Fri	Sat
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	1	2	3	4	5	6
7	8	9	10	11	12	13

Here the possible selection of dates is clearly visible. Clicking on a date will change the input field to that date.

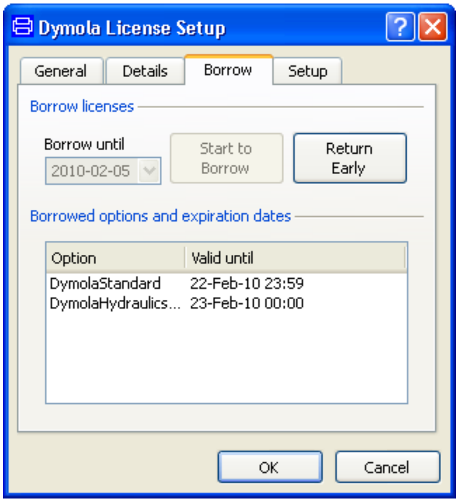
3. Click on **Start to Borrow**. The following message will appear:



Borrowing has been initiated. Restart Dymola and check out all options you need before disconnecting from the license server.

4. Click **OK** and **OK** and restart Dymola (while still connected to the server); now the basic borrowing is performed. (Borrowing will be indicated in several ways, please see next section.)

5. Open all libraries/options that you will need during your borrowing time. This will ensure that the appropriate license features are stored locally. The list in the lower half of the dialog displays currently borrowed licenses and when they will be automatically returned to the server.



Borrow until 2010-02-05

Option	Valid until
DymolaStandard	22-Feb-10 23:59
DymolaHydraulics...	23-Feb-10 00:00

In this example the Hydraulics library was opened; DymolaStandard indicates borrowing of Dymola without any options.

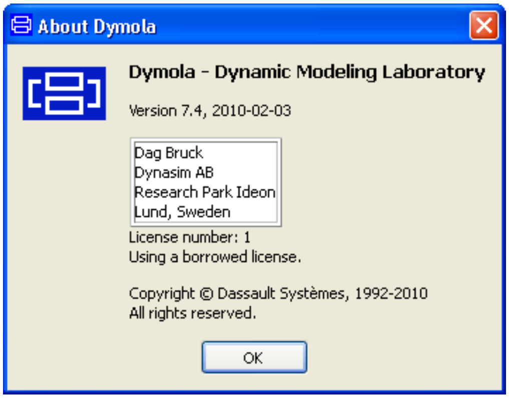
Please be careful not to open libraries/options that might be needed for others unless you really intend to do so. (Borrowing an option only available for one user only might not be appreciated by others.)

6. Finally disconnect from the license server **while Dymola is still running**. This step will create the local license file with the borrowed license. After disconnecting Dymola can be stopped.

Running Dymola

During the borrowing period Dymola can be started and stopped as often as needed. When license borrowing is used, Dymola displays it on the splash screen shown when starting Dymola and when using the command **Help > About Dymola**:

Borrowing period in About dialog.



About Dymola

Dymola - Dynamic Modeling Laboratory

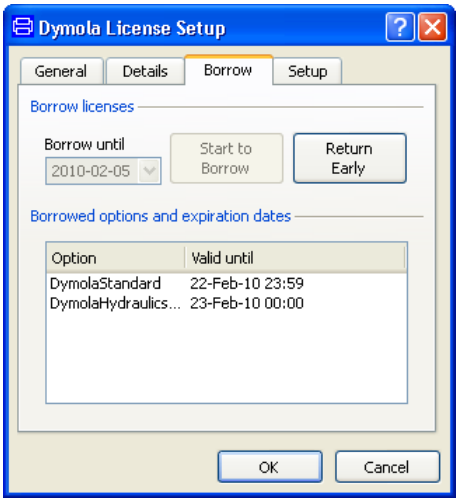
Version 7.4, 2010-02-03

Dag Bruck
Dynasim AB
Research Park Ideon
Lund, Sweden

License number: 1
Using a borrowed license.

Copyright © Dassault Systèmes, 1992-2010
All rights reserved.

Most information is given using the command **Help > License…**, in the **Borrow** tab.



Dymola License Setup

Borrow licenses

Borrow until: 2010-02-05

Borrowed options and expiration dates

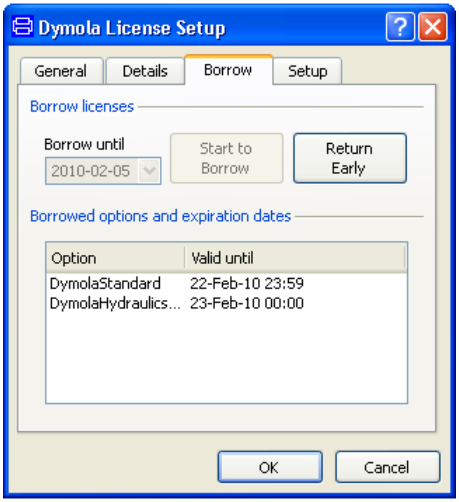
Option	Valid until
DymolaStandard	22-Feb-10 23:59
DymolaHydraulics…	23-Feb-10 00:00

Returning a license before expiration of borrowing (early return)

Currently borrowed licenses can be returned early when the computer is connected to the license server again.

In order to do an early return, do the following:

While Dymola is connected to the server, use the command **Help > License…**, and select the **Borrow tab**.



Borrow until 2010-02-05

Option	Valid until
DymolaStandard	22-Feb-10 23:59
DymolaHydraulics...	23-Feb-10 00:00

Now click on **Return Early**. The license (including all listed options) is returned o the server. Next time Dymola is restarted, the license is checked out the usual way.

It is a good idea to check e.g. the splash screen when starting up to convince oneself that the return was successful (in that case borrowing will not be mentioned in the splash screen).

A license returned to the license server cannot be checked out again until after approximately 2 minutes. If licenses are returned by e.g. exiting Dymola, but Dymola is restarted within approximately 2 minutes, the return is never performed.

License server options file

FLEXnet include tools for the local administrator. The options file allows the license administrator to control various operating parameters of the Dymola license server.

For example, it allows the administrator to

- Allow or deny the use of options by users.
- Reserve licenses for specified users.
- Control how many licenses can be borrowed and for how long.

The options file shall be called `dynasim.opt` and placed in the same directory as the Dymola license file of the license server.

An example of an options file that reserves a Dymola + Hydraulics library license for the user Bob is

```
RESERVE 1 DymolaStandard USER Bob
RESERVE 1 DymolaHydraulics USER Bob
```

Applicable “feature” and user names can be found in the license server log file. The details of the options file are described in Chapter 5 of “FLEXnet Licensing End User Guide”, which is available on request.

1.4 Dymola License Server on Linux

This section refers only to the Linux version of Dymola.

This section covers Linux-specific parts of the Dymola license server. For general items, e.g. background and how to set up the server using `lmtools.exe`, please see corresponding section on Dymola License Server on Windows.

Note! Dymola requires support of FLEXnet Publisher version 11.11. This version is part of the Dymola distribution for Linux.

The Linux license server for Dymola is located in a separate tar file.

To start the server the dynasim and the lm* files need to be installed, for example in `/usr/local/bin`. The server is started with the command

```
lmgrd –c<path to license file> -l<path to logfile>
```

A check with `pg aguxf` should show two new processes, `lmgrd` and `dynasim`. The server status can be checked with `lmutil lmstat –a`. In case of problems the log file should be examined.

To start the license server automatically when the system is rebooted, please update e.g. `/etc/rc.d/rc.local` accordingly. Note that the license server needs not to run as “root”.

Note. The license server by default uses the ports 27000-27009. They can be configured if needed, e.g. if there are issues with firewalls.

Full details of FLEXnet license server installation can be found in the FLEXnet User’s Manual, which can be downloaded from www.flexera.com.

License borrowing on Linux

License borrowing is enabled by setting the environment variable LM_BORROW. The value must specify beginning and end dates of the borrowing period, as well as the vendor name “dynasim”. The general format is:

```
LM_BORROW=<start date>:dynasim:<end date>
```

An example (using bash) which specifies the start date 10 November 2009 and the end date 12 November 2009 is:

```
export LM_BORROW=10-nov-2009:dynasim:12-nov-2009
```

After setting the environment variable LM_BORROW, Dymola must be restarted and the appropriate license options checked out before disconnecting from the license server.

The status of borrowing can be displayed in the Linux server using a status command. An example, for a 64 bit Dymola 2016, is:

```
/opt/dymola-2016-x86_64/bin/lmutil lmborrow –status
```

The command displays the names of borrowed features and the expiration dates.

Returning a license before expiration of borrowing (early return)

Currently borrowed licenses can be returned early when the computer is connected to the license server again.

The names of the features that are currently borrowed can be seen using the status command in the previous section. When returning, any of these names must be used in the return command below.

In order to do an early return, give a return command while Dymola is connected to the server. An example returning the license for Pneumatics Library, for a 64 bit Dymola 2016, is:

```
/opt/dymola-2016-x86_64/bin/lmutil lmborrow –return –c ~/.dynasim/
    dymola.lic DymolaPneumaticsLib
```

Whether the return was made can be seen using the status command in previous section.

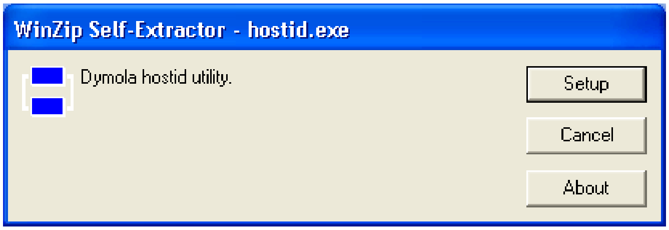
A license returned to the license server cannot be checked out again until after approximately 2 minutes. If licenses are returned by e.g. exiting Dymola, but Dymola is restarted within approximately 2 minutes, the return is never performed.

1.5 Utility programs

1.5.1 Obtaining a host id

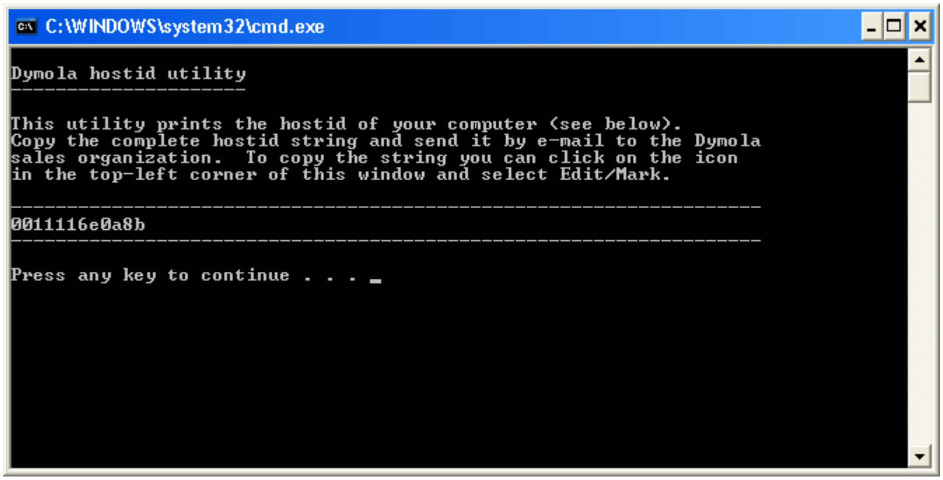
To be able to easy find out the host id of a computer without having Dymola installed, a small file `hostid.exe` can be obtained from your Dymola distributor. (If Dymola demo is installed, the host id can also be found using Dymola, please see section “Obtaining a host id” on page 15.)

Executing this file (by double-clicking it or opening it) the following menu will be displayed:


WinZip Self-Extractor - hostid.exe

Dymola hostid utility.

Selecting **Setup** will display the following:



```
Dymola hostid utility
---------------------

This utility prints the hostid of your computer (see below).
Copy the complete hostid string and send it by e-mail to the Dymola
sales organization.  To copy the string you can click on the icon
in the top-left corner of this window and select Edit/Mark.

-------------------------------------------------------------------
0011116e0a8b
-------------------------------------------------------------------

Press any key to continue . . . _
```

Clicking in the upper left corner and selecting **Edit > Mark** makes it possible to selecting the host id by dragging the cursor over it. Once selected, **Edit > Copy** will place the host id in the clipboard, from where it should be pasted into a mail to your Dymola distributor.

1.6 System requirements

1.6.1 Hardware requirements

- At least 1 GB RAM
- At least 400 MB disc space

1.6.2 Hardware recommendations

At present, it is recommended to have a system with an Intel Core 2 Duo processor or better, with at least 2 MB of L2 cache. Memory speed and cache size are key parameters to achieve maximum simulation performance.

A dual processor will be enough if not using multi-core support; the simulation itself, by default, uses only one execution thread so there is no need for a "quad" processor. If using multi-core support, you might want to use more processors/cores.

Memory size may be significant for translating big models and plotting large result files, but the simulation itself does not require so much memory. Recommended memory size is 4 GB of RAM for 32-bit architecture and 6 GB of RAM for 64-bit architecture.

1.6.3 Software requirements

Microsoft Windows

Dymola versions on Windows and Windows operating systems versions

Dymola 2016 is supported, as 32 or 64 bit application, on Microsoft Windows 7 and Windows 8.1. Since Dymola does not use any features supported only by specific editions of Windows ("Home", "Professional", "Enterprise" etc.); all such editions are thus supported if the main version is supported.

Compilers

Please note that for the Windows platform, a Microsoft C/C++ compiler, or a GCC compiler, must be installed separately. The following compilers are supported for Dymola 2016 on Windows:

Visual Studio - Free editions:

- Visual Studio 2008 Express Edition (9.0)
- Visual C++ 2010 Express (10.0)
- Visual Studio 2012 Express Edition (11.0)
- Visual Studio 2013 Express Edition (12.0)

Visual Studio - Professional editions:

- Visual Studio 2005 (8.0)
- Visual Studio 2008 (9.0)
- Visual Studio 2010 (10.0)
- Visual Studio 2012 (11.0)
- Visual Studio 2013 (12.0)

GCC compilers

- 32-bit MinGW, with a GCC version compatible with 4.8.1.
- 64-bit MinGW, with a GCC version compatible with 4.9.2.

Note – the GCC compilers have some limitations, and demand for an add-ons etc, please see section "GCC compiler" on page 11.

Dymola license server

For a Dymola license server on Windows, all files needed to set up and run a Dymola license server on Windows, except the license file, are available in the Dymola distribution. (This includes also the license daemon, where Dymola presently supports FLEXnet Publisher version 11.11. A recent version is part of the Dymola distribution.)

Linux

Supported Linux versions and compilers

Dymola 2016 is supported on SUSE Linux (Release 11), 32-bit and 64-bit, with gcc version 4.3.4, and compatible systems.

In addition to gcc, the model C code can also be compiled by clang. To change compiler, change the variable `CC` in

```
/opt/dymola-2016-<architecture>/insert/dsbuild.sh
```

`<architecture>` is `i586` for 32-bit application, `x86_64` for 64 bit-application; for example for the 64-bit application the path is:

```
/opt/dymola-2016-x86_64/insert/dsbuild.sh
```

Dymola 2016 is supported as a 32-bit and 64 bit application on Linux.

Notes

- For rendering of jpg files, libjpg62 must be installed.
- 32-bit compilation might require explicit installation of 32-bit libc. E.g. on Ubuntu: `sudo apt-get install g++-multilib libc6-dev-386`

Note on libraries

- Please note that you have to use the Optimization library version 2.x or higher to use multi-criteria design optimization on Linux; the older Design.Optimization package does not support multi-criteria design optimization on Linux.
- The library UserInteraction is not supported on Linux.

Dymola license server

For a Dymola license server on Linux, all files needed to set up and run a Dymola license server on Linux, except the license file, are available in the Dymola distribution. (This includes also the license daemon, where Dymola presently supports FLEXnet Publisher version 11.11. This version is part of the Dymola distribution.)

1.7 License requirements

1.7.1 General

This description covers the license requirements for different features; for description of the features themselves, please see relevant documentation in “Dymola User Manual Volume 2”, chapter “Other simulation environments”, and the documentation for the libraries.

This description assumes that machines having Dymola installed, have the Dymola **Standard Configuration** license.

Apart from the **Standard Configuration** license, there are other licenses for different categories of users:

- **Demo** license
- Licenses for academia (**Academic Learn/Innovate**, **Student Learn/Innovate**)

Some limitations may apply to those; in particular the demo license is limited.

Licenses can be node-locked (to a certain machine) or sharable (accessible from a license server). Note that node-locked and sharable licenses cannot be mixed.

1.7.2 License for Dymola – Simulink interface

To use the traditional connection between Dymola and Simulink (Dymola-on-Windows Mode), you need a **Simulink Interface** license in Dymola.

Note that this mode is not available for Linux. For Linux, see section "Import to Simulink"; the second part.

1.7.3 License for real-time simulation

In Dymola

To be able to enable inline integration, needed for real-time simulation, you must have the **Real-Time Simulation** license. (The **Source Code Generation** license includes this functionality as well.)

On dSPACE and xPC Target platforms

To be able to use real-time simulation on dSPACE or xPC Target platforms, you must have

- The **Simulink Interface** license in Dymola to use the Dymola-Simulink interface.
- Any of the licenses **Real-Time Simulation** or **Source Code Generation**, depending on what you want to do:
    - o If you only want to use models that use inline integration, the **Real-Time Simulation** license is sufficient.
    - o If you want to use other models as well, you need the **Source Code Generation** license.

1.7.4 Licenses for exporting code

The assumption is that the model is to be imported/executed on another machine than the one where it was created.

Exporting a Dymola model to a machine without a Dymola license

See the corresponding import section below, that also explains the export.

Exporting a model to Simulink

In Windows you can use the “Dymola-on-Windows” mode of the Dymola-Simulink interface (using DDE communication between Dymola and Simulink). In this mode, you can export a compiled model and execute it on another Windows computer. You don´t need any license to export the compiled model, but there might be requirements depending on how you want to use it – see the section about importing a model to Simulink below.

For other platforms (for example Linux) – but also optionally in Windows – you can use the “Import” mode of the Dymola-Simulink interface. You can export models to be imported in this mode. To do this, you must use the ExternalInterfaces library, the `ExternalInterfaces.Export.toSimulink` function. You need the **Simulink Interface** license to do this. Depending on how the model should be used, there are additional requirements, see section “Importing a model to Simulink” below.

Exporting a model for integration with Real-Time Workshop (dSPACE and xPC)

You can do this by using the ExternalInterfaces library, the `ExternalInterfaces.Export.toSimulink` function. You need to have both the **Simulink Interface** license, and the **Real-Time Simulation** license to do such an export. (If you have a **Source Code Generation** license, that one includes **Real-Time Simulation** as well.)

Exporting a model to FMU format

See the section “Importing an FMU” below; this section also explains the export.

Exporting a model to Binary

You can export a model to binary. This requires the license **Binary Model Export**. (The **Source Code Generation** license includes this functionality as well.) Such an exported model can be executed on a machine without having any Dymola license.

Exporting a model to Source Code (C code)

You can export a model to source code. This requires the license **Source Code Generation**. Such an exported model can be executed on a machine without having any Dymola license. Additional features are available, see the chapter “Other simulation environments”.

1.7.5 Licenses for executing/importing/using code

Executing a Dymola model on a machine without a Dymola license

There are two cases:

- The model has been generated by a machine having a **Binary Model Export** license (or **Source Code Generation** license). In this case no Dymola license is required to execute the model.

- The model has been generated without any of the two licenses above. You can execute the model if the machine has a **Standard Runtime Configuration** license, and the environment variable DYMOLA_RUNTIME_LICENSE is set to the path where the license is located. (This is needed for the executable model to find the license file.)

Import to Simulink

If you have used the “Dymola-On-Windows” mode to export a compiled model, you can use it in a number of ways:

- You can execute the model on a Windows computer without Dymola installation in two cases:
    - The model has been generated by a machine having a **Binary Model Export** license (or **Source Code Generation** license). In this case no Dymola license is required to execute the model.
    - The model has been generated without any of the two licenses above. You can execute the model if the machine has a **Standard Runtime Configuration** license, and the environment variable DYMOLA_RUNTIME_LICENSE is set to the path where the license is located.
- You can import it in Simulink using the “Import” mode of the Dymola-Simulink interface. This requires that the model has been generated by a machine having a **Binary Model Export** license (or **Source Code Generation** license).

If you have exported a model using the using the ExternalInterfaces library, the `ExternalInterfaces.Export.toSimulink` function, you can import it using the “Import” mode of the Dymola-Simulink interface. Such an imported model can be compiled, simulated, and also downloaded to real-time platforms (the last option requires that it has been exported from a machine having the **Real-Time Simulation** license, or the **Source Code Generation** license).

Importing an FMU

You have three cases:

- The FMU is a Dymola FMU, and generated by a machine having a **Binary Model Export** license (or **Source Code Generation** license); in this case you can import it and execute it on a computer without a Dymola license.
- The FMU is a Dymola FMU, and generated by a machine without any of the licenses above; in this case you must fulfill two conditions to be able to execute the FMU:
    - The machine you import to must have a **Standard Runtime Configuration** license.
    - The environment variable DYMOLA_RUNTIME_LICENSE must be set to the path where the license is located. (This is needed for the dll of the FMU to find the license file.)

Important! If you have a Dymola license on the machine, the environment variable must still be set.

- The FMU is from another vendor. For this case, please see the documentation for that vendor; there are no additional requirements from Dymola.

1.7.6 Licenses for libraries in the Dymola library menu

A number of libraries in the Dymola library menu can be used in Dymola without any additional licenses, a number of libraries require additional licenses when working with more advanced features, and a number of libraries always require additional licenses.

Note:

- If you want to export code, the above requirements for export licenses apply as well.

Free libraries and packages in the Dymola library menu

- Automotive Demos (in the menu **File > Demos**)
- DataFiles
- Design.Experimentation
- Design.Validation
- DymolaCommands
- Modelica Reference
- Modelica Standard Library
- Modelica_DeviceDrivers
- Modelica_LinearSystems2
- Modelica_StateGraph2
- Modelica_Synchronous
- Plot 3D
- UserInteraction
- VehicleInterfaces

Libraries in Dymola library menu that require additional licenses when for example running more advanced examples

- Design.Calibration – requires **Model Calibration** license when running more than one tuner.
- Design.Optimization – requires **Design Optimization** license when running more advanced examples. Note that there is a newer library available for optimization.

Libraries in Dymola library menu always requiring additional licenses

Libraries in the Dymola library menu not listed above always require additional licenses.

1.7.7 Licenses for libraries not in Dymola library menu

There are a number of libraries not in Dymola library menu, for example:

- Libraries can be downloaded from the homepage of Modelica Association (www.Modelica.org) – some are free, some require license.
- Third party vendors can use the license handling in Modelica to add license requirements on their libraries.

1.8 Troubleshooting

This is a common section for both Windows and Linux. If a problem only is applicable for e.g. Linux, it is stated.

Occasionally the installation will not succeed, or the program will not operate as intended after installation. This section will outline some of the problems that have been detected in the past.

1.8.1 License file

The license file used is not the one wanted

There are a number of standard paths where Dymola searches for a valid license. In an old invalid license is stored by mistake in one of those locations, that license might be tried instead of the correct one. Information about which license is currently in use by Dymola is given using the command **Help > License > Setup**. The path to that license is specified by `Filename` in that tab.

License file is not authentic

The error message “License file not authentic” indicates either an error in the license file, or a mismatch between your computer system and your license file.

- The license file is locked to your computer system, which means that you cannot execute Dymola from another computer.
- The license file format has been changed in Dymola 7.0 and later versions. If you also have older versions of Dymola installed, please check that you have a new license file as well.

Additional information

If there is some error in the license file or with the license server, Dymola presents a short error message by default. A more detailed description, including FLEXnet Publisher error codes, is produced if Dymola is started with the command line option `/FLEXlmDiag`. On Windows, start a command (DOS) window (using the command **Start > All Programs > Accessories > Command Prompt** in Windows) and issue the following commands (assuming Dymola 2016 64-bit is used on a 64-bit computer):

```
cd \Program Files (x86)\Dymola 2016\bin64
dymola.exe /FLEXlmDiag
```

On Linux the command will be:

```
dymola /FLEXlmDiag
```

The additional information will in many cases be helpful in correspondence with support.

License server

Correct operation of the license server should be verified with `lmtools.exe`, see "Installing the license server" on page 31. The FLEXnet Publisher logfile provides additional information about the day-to-day operation of the server.

Always using the latest version of the FLEXnet Publisher license daemon `lmgrd.exe` is strongly recommended. It is guaranteed to be compatible with all earlier versions of FLEXnet Publisher.

License borrowing

Different versions of Dymola

There are limitations regarding license borrowing when borrowing is done in one version of Dymola, and using the borrowed license is used in another version of Dymola on the same PC.

For Windows, a license borrowed using Dymola 7.4 FD01 or older cannot be used by Dymola 2012 or newer without being connected to the license server.

For Linux, a license borrowed using Dymola 2012 FD01 or older cannot be used by Dymola 2013 or newer without being connected to the license server.

If access to e. g. both Dymola 7.4 FD01 and Dymola 2012 is required on a Windows PC, both versions must be used to borrow, by the following procedure:

- Initiate borrowing with any Dymola version.
- Open Dymola 7.4 FD01 (or older) and check out the required features.
- Open Dymola 2012 (or newer) and check out the required features.
- Validate by checking that there are two entries of all the required features in the **Details** tab, using the command **Help > License…**.
- Disconnect from the network and validate that both versions can be run as expected.

License borrowing of 32/64-bit Dymola

When borrowing license, only the license of the Dymola version you run will be borrowed: and 64-bit and 32-bit Dymola are seen as different versions. For the few cases when you want to

- Borrow a license, AND
- run Dymosim.exe outside of Dymola, AND
- do not have export option;

we advice that you in 64-bit Dymola generate a 64-bit Dymosim.exe using the flag `Advanced.CompileWith64=2`. For more information about this flag, see chapter “Simulating a model”, section “Dynamic Model Simulator”.

Sharable licenses

Please note that if a new session is started in Windows by using **Log Off > Switch User** the original user is still logged on and any Dymola program occupies a sharable license.

1.8.2 Compiler problems

Test compiler button and error messages

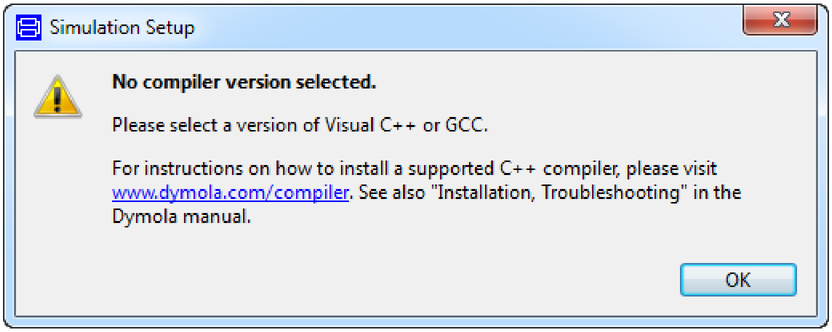
The compiler used to compile the C code generated by Dymola into executable code for simulation is set in the **Compiler** tab using the command **Simulation > Setup…**, see chapter “Simulating a model”, section “Editor command reference – Simulation mode”, sub-section “Main window: Simulation menu”, command “Simulation > Setup…”.

Some potential problems can be found by pressing the **Test compiler** button in the Compiler tab (see above). Any warning messages indicate problems that need to be resolved before translating a model. Pressing the button performs a number of tests:

- Validates the DOS environment.
- Check the Dymola installation for run-time libraries.
- Verifies that the selected compiler directory contains a valid compiler.
- Validates that the compiler can compile by executing a small test model.

The compiler is tested for both 32-bit and 64-bit mode.

Error messages with information how to proceed (including a link to the web page described below) will be displayed, e. g. when no compiler is selected:

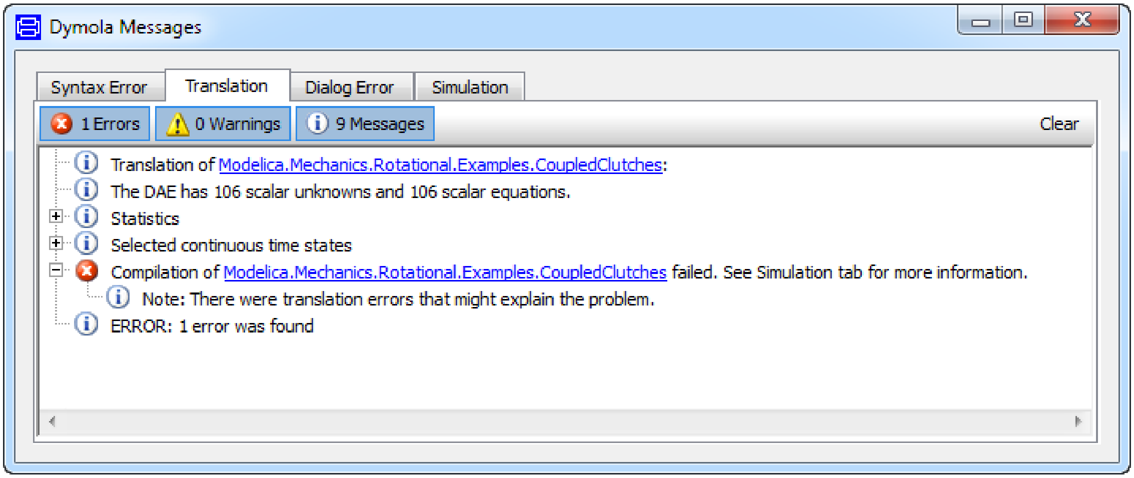


No compiler version selected.

Please select a version of Visual C++ or GCC.

For instructions on how to install a supported C++ compiler, please visit www.dymola.com/compiler. See also "Installation, Troubleshooting" in the Dymola manual.

If no compiler has been selected (or installed), the corresponding information will also be displayed in the command log, in the **Translation** tab:



Translation of Modelica.Mechanics.Rotational.Examples.CoupledClutches:

The DAE has 106 scalar unknowns and 106 scalar equations.

Statistics

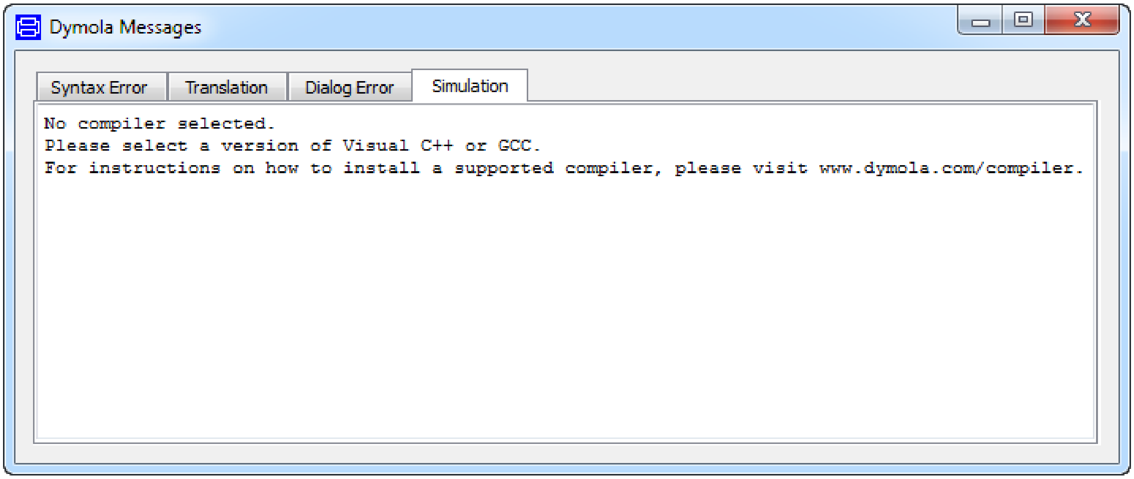
Selected continuous time states

Compilation of Modelica.Mechanics.Rotational.Examples.CoupledClutches failed. See Simulation tab for more information.

Note: There were translation errors that might explain the problem.

ERROR: 1 error was found

And also in the **Simulation** tab:



```
No compiler selected.
Please select a version of Visual C++ or GCC.
For instructions on how to install a supported compiler, please visit www.dymola.com/compiler.
```

Dymola webpage for compiler issues

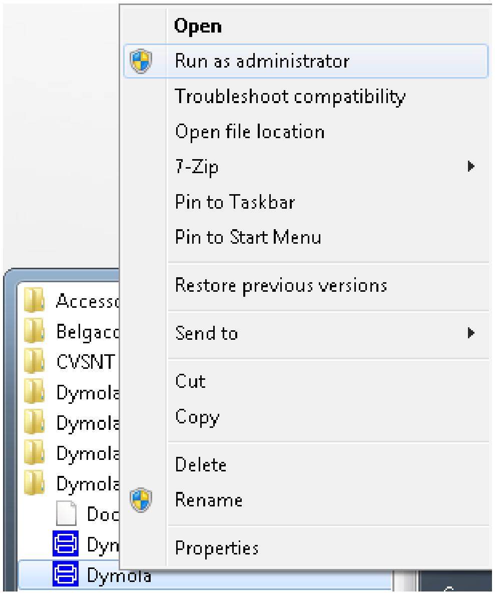
A web page is available for Dymola compiler issues: www.Dymola.com/compiler. This page contains the following information:

- Links to download compilers.
- Extract from the documentation concerning installation and troubleshooting of compilers.

A link to this page is presented when detecting any error using the **Find compiler** button and in the build log when no compiler is installed/selected.

Dymola on Windows 7

Note that Visual Studio performs parts of its installation the first time it is run. This must be performed in order to use Visual Studio with Dymola, a step that requires administrator privileges. The rights can be elevated by running Dymola as administrator the first time. This is done by right-clicking the Dymola icon is the Windows Start menu and selecting **Run as administrator**.



To validate this, any model (e. g. a demo) should be opened and translated in Dymola.

1.8.3 Simulink

If the Dymola-Simulink interface does not work, please check the following (some of which may sound elementary):

- You have a Dymola license that supports the Simulink interface. Note that Simulink support is a separate option.

- You have included the three directories `Dymola 2016\mfiles`, `Dymola 2016\mfiles\traj` and `Dymola 2016\mfiles\dymtools` in the Matlab path. These have to be included every time you want to use the Dymola-Simulink interface and it is a good idea to store the included paths in Matlab.
- You can find the interface in Simulink's browser as `Dymola Block/DymolaBlock` (if not, you have probably not included the directories, mentioned above, into the Matlab path).
- Make sure that you have a Visual Studio C++ compiler installed on your computer. Make sure that the Matlab mex utility has been configured to use that compiler (type `mex –setup` in Matlab to configure). Finally, test by trying to compile and link an example mex file, e.g. `matlab\extern\examples\mex\yprime.c`.
- You have created external inputs to the Dymola Block, and outputs from the Dymola Block, in a correct way. See also the manual “Dymola User Manual Volume 2”, chapter “Other Simulation Environments”, section “Using the Dymola-Simulink interface”, sub-section “Graphical interface between Simulink and Dymola”.
- You have compiled all Dymola models used in the model; otherwise you will get an error message.
- If “Allow multiple copies of block” is unchecked you should not copy the block. Unchecking it should only be done if you have a dSPACE system.

Also note that the parameterizations differ between blocks in the Modelica Standard Library and in Simulink. For example, the frequency of Simulink's Sine-block is measured in rad/s, which is commonly known as angular frequency and should thus be 2π times the frequency in the corresponding source in Modelica.

Only Visual Studio C++ compilers are supported to generate the DymolaBlock S-function. The LCC compiler is not supported.

1.8.4 Change of language

Dymola is available in Japanese. Sometimes the user wants to change the language after installation. Please see section “Language” on page 18 on how to do this.

1.8.5 Other Windows-related problems

Starting the installation

The installation normally starts automatically when you insert the distribution DVD. If auto start has been disabled, please start `D:\setup.exe` (assuming your DVD drive is labeled D) from Windows Explorer by double-clicking on the file or use the **Start** button in Windows, select **Run**, enter `D:\setup.exe` and click **OK**.

Deep directory hierarchies

Compilation and simulation of the model may fail in a very deep directory hierarchy, if the length of the directory path exceeds 250 characters. This is caused by a bug in Microsoft software, and we are investigating ways to work around it.

Writable root directory

Due to a bug in some versions of the Microsoft runtime library, the root directory C:\ should be writable in order to store temporary files. If that is not the case, Dymola will create working files in the current directory, which are not automatically deleted.

2 Index

A

AddModelicaPath, 21

B

borrowing
  general, 35
  on Linux, 40
  on Windows, 35

C

C compiler, 11
compiler, 11
  GCC (on Linux), 28
  GCC (on Windows), 11
  installation, 11
  Microsoft Visual C++, 11
  troubleshooting, 51

D

daemon
  license, 31
  vendor, 31
Dymola
  32-bit application on Linux, 28
  32-bit application on Windows, 6
  64-bit application on Windows, 6
Dymola license server
  on Linux, 40
  on Windows, 31

E

environment variables
  DYMOLA, 29
  DYMOLA_RUNTIME_LICENSE, 17
  DYMOLAPATH, 29
  DYMOLAWORK, 8, 29
  LM_BORROW, 40
  MODELICAPATH, 21, 29

F

file extensions
  .cab, 19
  .msi, 19
FLEXnet, 31

license server options file, 39

G

GCC (on Linux), 28
GCC (on Windows), 11

H

host id, 15

I

incorrect license file format, 18
installation
- environment variables, 29
- license daemon, 31
- license server on Linux, 40
- license server on Windows, 31
- Linux, 28
- remote on Windows, 19
- troubleshooting, 49
- windows, 6

L

language setting, 18
library
- Modelica 2-compatible libraries, 20

libraryinfo.mos, 21
license
- borrowing - general, 35
- borrowing - on Linux, 40
- borrowing - on Windows, 35
- daemon, 31
- demands, 44
- early return - on Linux, 41
- early return – on Windows, 38
- error message, 49
- expiration - settings, 23
- incorrect file format, 18
- node-locked, 15
- not found or invalid - settings, 23
- preventing checking out - in Linux, 30
- preventing checking out – in Windows, 23
- requirements, 44
- runtime, 17
- server
  - on Linux, 40
  - on Windows, 31
- server options file, 39
- sharable, 14

Linux
- Dymola as 32-bit application, 28
- Dymola license server, 40
- installation, 28

local host id, 15

M

Microsoft
- Visual C++, 11
- Windows installation, 6

Modelica 2-compatible libraries, 20
MODELICAPATH, 21
My Documents, 8

N

node-locked license, 15

P

Program Files, 8

R

runtime license, 17

S

sharable licenses, 14
Simulink
- troubleshooting, 53

system requirements
- hardware, minimum, 42
- hardware, recommended, 42
- software, Linux, 44
- software, Windows, 43

T

trobleshooting
- Dymola on Windows 7, 53

troubleshooting
- compiler, 51
- language, 54
- license borrowing, 50
- license file, 49
- license server, 50
- other windows problems, 54

Simulink, 53

U

UNC, 8
upgrading Dymola, 18

W

Windows
Dymola as 32-bit application, 6
Dymola as 64-bit application, 6
Dymola license server, 31
Windows 7
Dymola on Windows 7, 53
working directory
Dymola, 8
set default for Dymola, 19