OPAL-RT TECHNOLOGIES

FROM IMAGINATION TO REAL-TIME

RT-LAB Solution for Real-Time Applications

OP101 : Getting started

RT-LAB Software

Training Services

Outline

1. ***Launch RT-LAB***
2. Add a Target
3. Project and Models
4. Build, Load & Execute

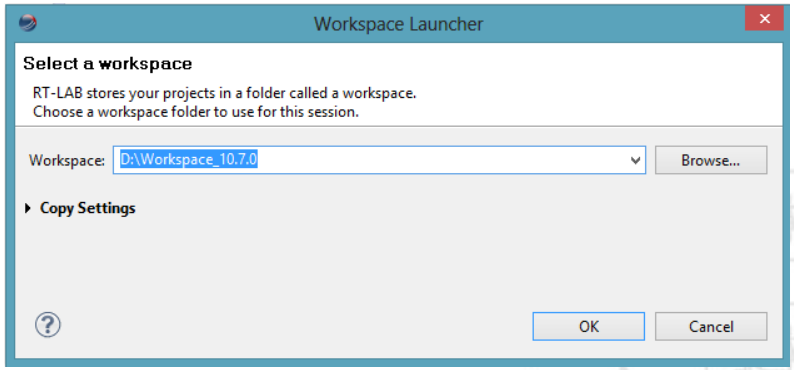
Launch RT-Lab

Basic information

- **MetaController**
  - Must be run as an administrator
  - Main application acting as a server
  - Manually opened *C:\OPAL-RT\RT-LAB\vxx.x.x.xxx\common\bin\MetaController.exe*
  - When running, available in the system tray

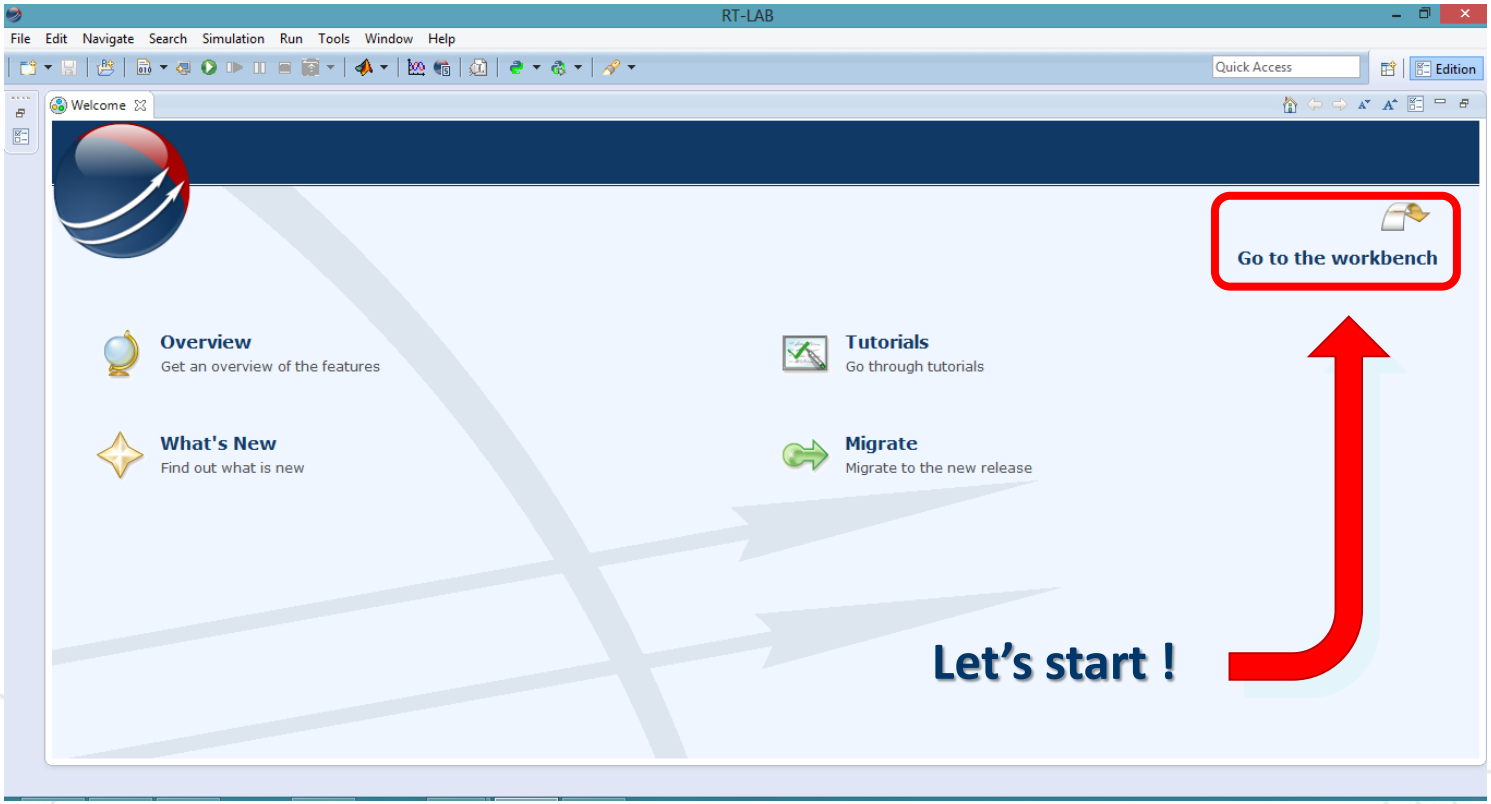
- **License**
  - Before RT-LAB **10.4**: associated to **host**'s MAC address
  - After RT-LAB **10.5**: associated to the **target**'s MAC address and Hard Disk Drive serial number

- **Workspace**
  - Directory where your work is stored
  - Can be changed anytime
    - → *File* → *Switch Workspace*



Launch RT-Lab

First Startup

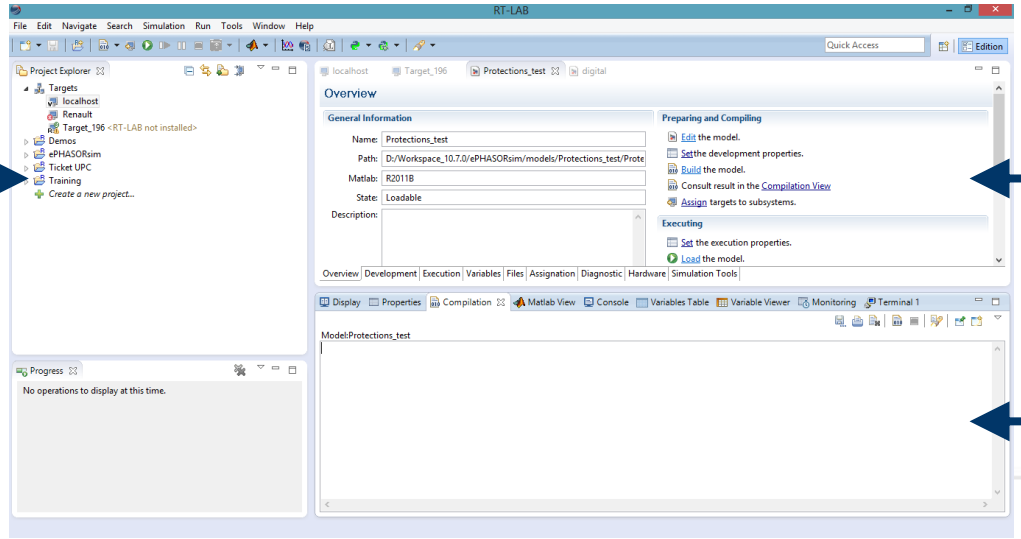


Launch RT-Lab

Workbench – Project Explorer, Editor & View

- **Project Explorer**
  - Hierarchical view of Targets, Projects, Models and Files
- **Editor**
  - Basic information about the selected Target, Model or File
  - Set parameters and interaction with the selected Target, Model or File
- **View**
  - Console and advanced properties of the active editor

Project Explorer



Editor

View

Outline

1. Launch RT-LAB

2. Add a Target

3. Project and Models

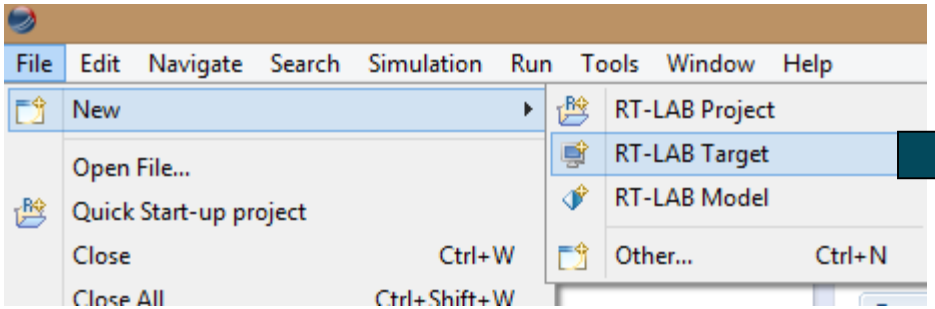
4. Build, Load & Execute

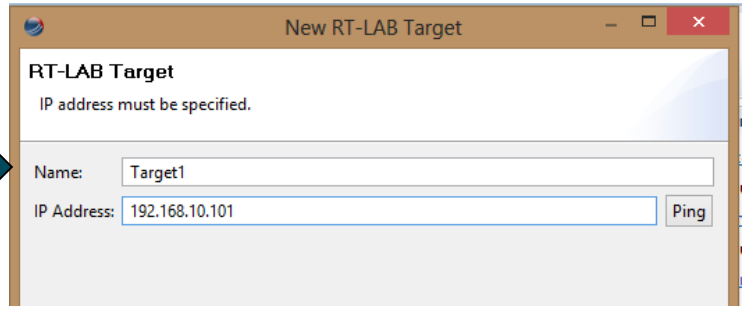
Add a Target

New target

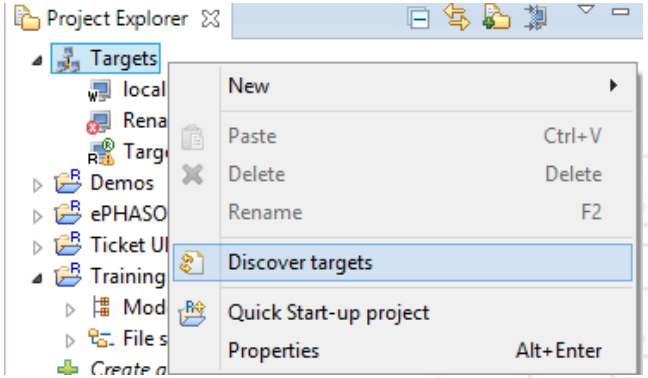
- **Requirements**
  - Target must be switched on and connected to the same network as the host PC

- **Using IP address → add Target manually**





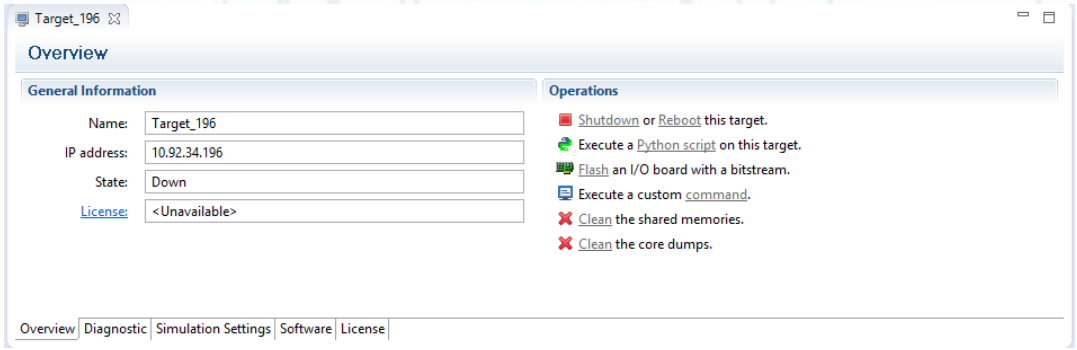
- **Discover targets -> add Target automatically**
  - RT-LAB will look for targets connected to the network

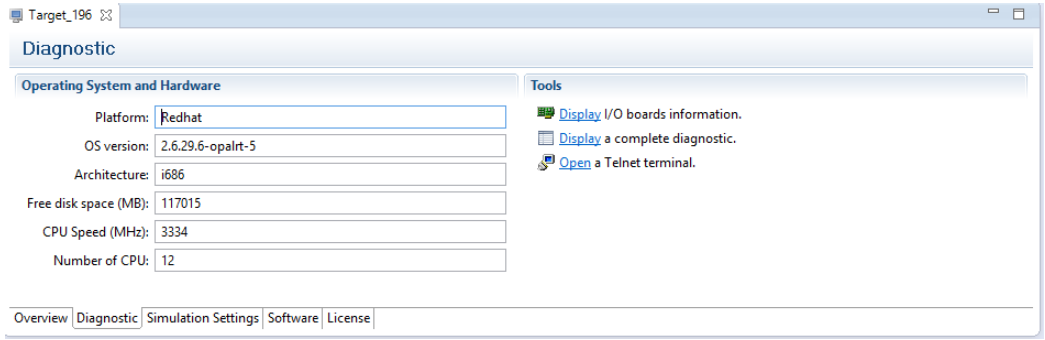


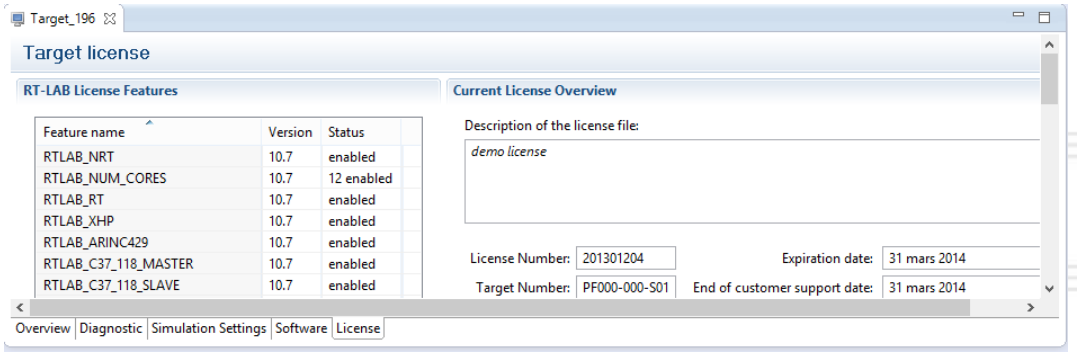
Add a Target

Edit target

- **Overview**
  - General Information
- **Diagnostic**
  - Operating System & Hardware
- **Simulation Settings**
  - Multimodel support
  - Date & Time
- **Software**
  - Install a new version
  - Set as default
- **License**
  - Install a new license
  - Request to technical support







Outline

1. Launch RT-LAB

2. Add a Target

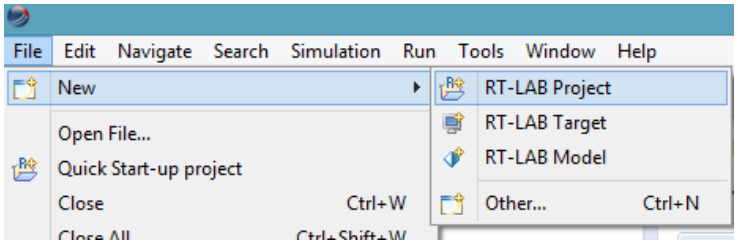
3. Project and Models

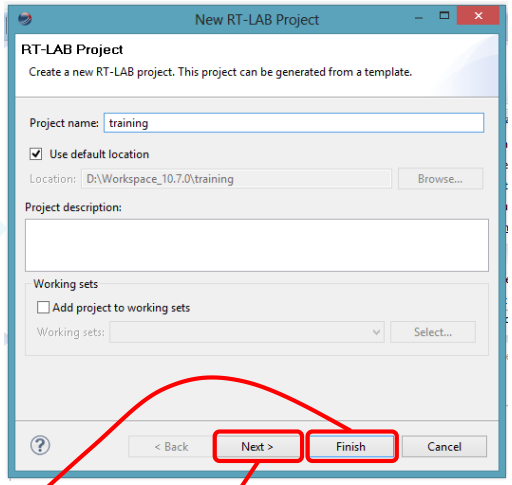
4. Build, Load & Execute

Projects and models

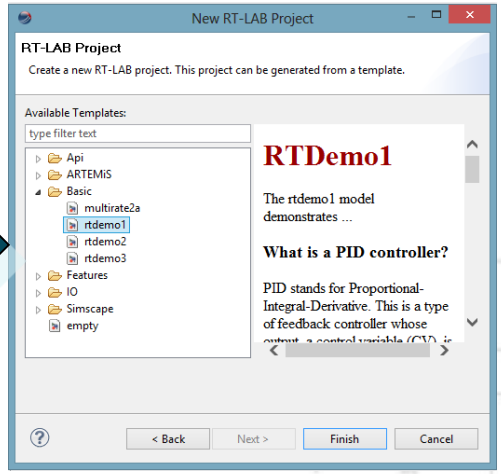
A project contains models and other resource files

- **New project**





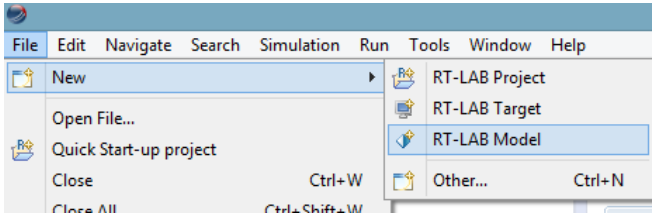
- **Add models inside a project**
  - Click on ***Finish*** to add your own models in the next step
  - Click on ***Next*** to add an example model located in "C:\OPAL-RT\RT-LAB\vxx.x.x.xxx\Examples" to the RT-LAB project

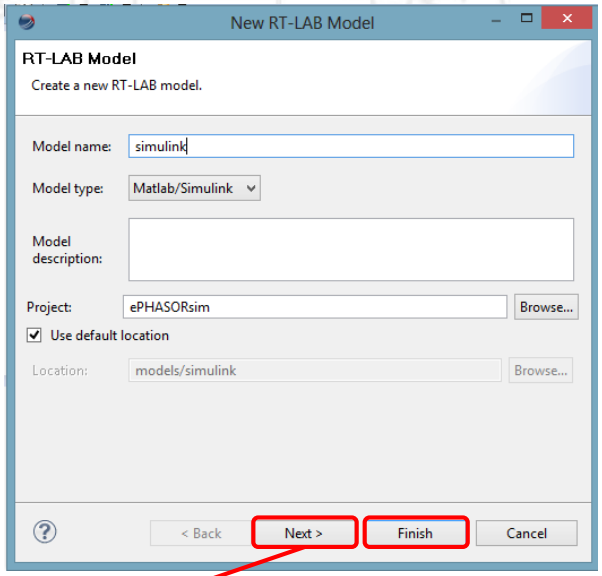


Projects and models

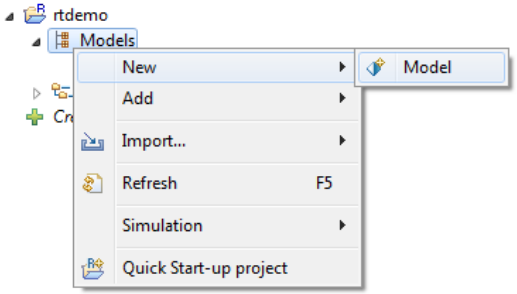
Create your own model

- **Model from scratch**

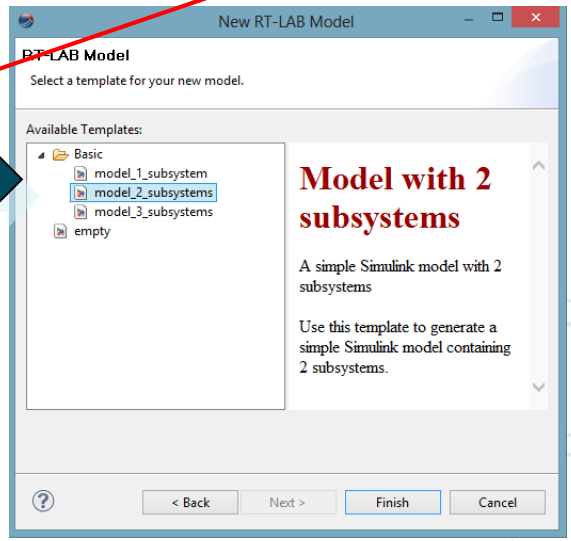




OR



- Click on ***Next*** to create a new model using a template file with subsystems



- Click on ***Finish*** to create an empty model

Projects and models

Create your own model

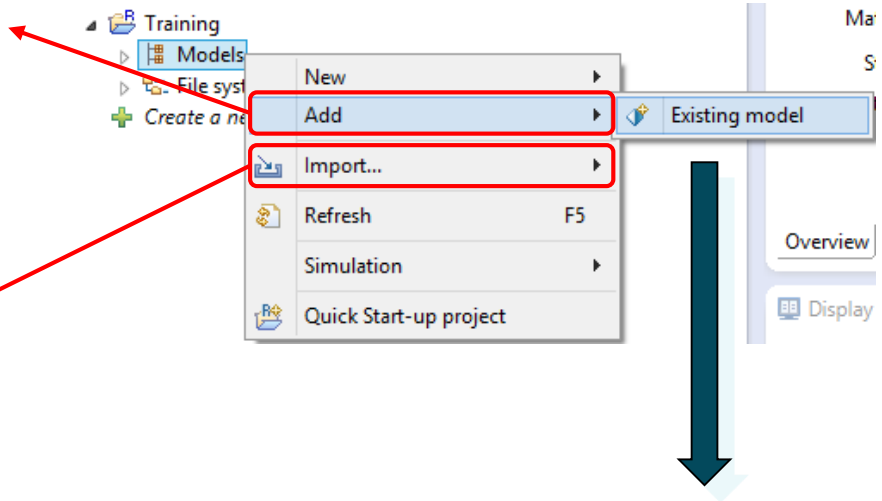
- **Existing model: import or link ?**

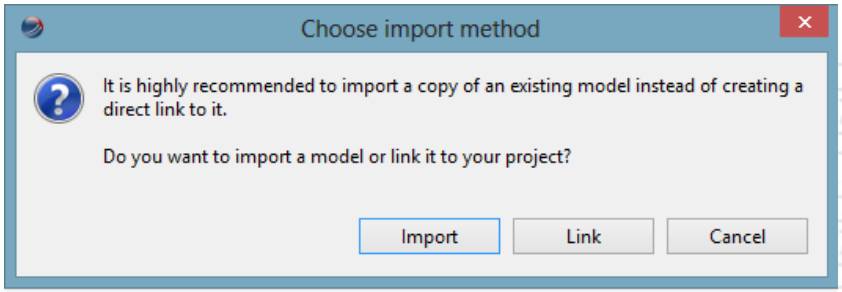
  - **Link (Add):** The model is not copied in the Workspace. A link to it is created.

    ➡ ***Original model is modified***

  - **Import:** create a ***copy*** of the original model inside the Workspace.

    ➡ ***Original model is not modified***





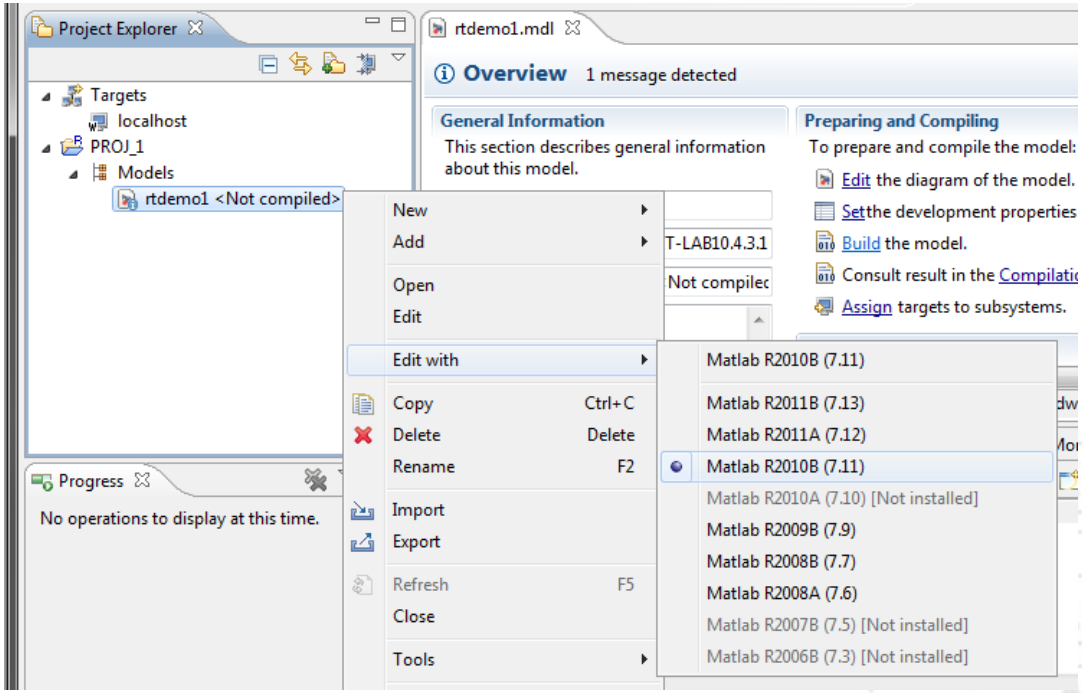
Projects and models

Edit a model

- **One can edit a Simulink model from MATLAB/Simulink environment (independently from RT-LAB)**

OR

- **Model can be opened from RT-LAB interface**
  - Right-clic on the model
  - Edit or Edit with…



Outline

1. Launch RT-LAB

2. Add a Target

3. Project and Models

4. Build, Load & Execute

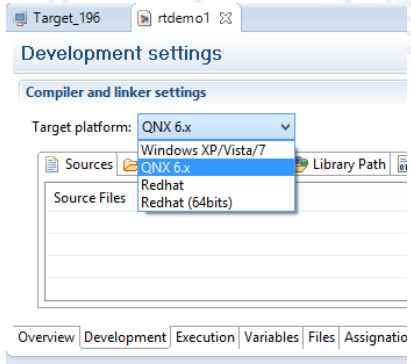
Build, Load & Execute

Build model

- **Information**
  - Software versions
  - Target platform (Windows, QNX or Redhat)

- **Model Separation**
  - One ****.mdl*** file per top-level subsystem (SM, SC and SS)

- **Generating C code**
  - By calling Real-Time Workshop (RTW) from MATLAB
  - Applied to each individual model




```
-------------------- Starting compilation --------------------
Start at : Thursday, September 24, 2009, 16:32:38

The current RT-LAB version is:  v8.3.3
The current model is:           C:\Opal-rt\RT-LAB8.3.3\Examples\Basic
The current host platform is:   XP/Vista
The current target platform is: QNX 6.x

Preparing original model for code separation and generation...

Matlab version v7.0.1 is different from model version v6.5
The current Matlab version is:  v7.0.1

Separating RT-LAB subsystem 'sc_user_interface'...
Separating RT-LAB subsystem 'sm_controller'...
Separating RT-LAB subsystem 'ss_plant'...

Model preparation and separation duration : 00h:00m:29s
-------------------- Completed successfully --------------------
```

Info

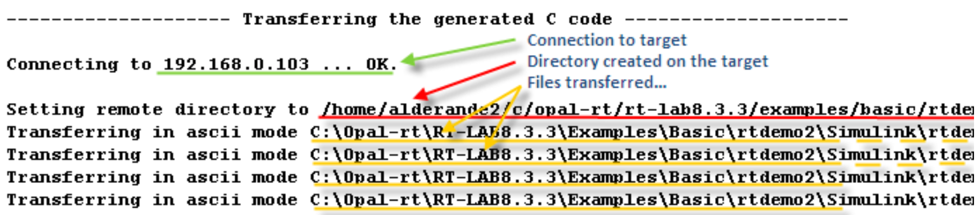
Separation

```
-------------------- Generating C code --------------------

Using System Target File (TLC file) : rtlab.tlc...

Using Template Makefile (TMF file) : rtlab.tmf...

-------------------- Generating rtdemo2_1_sm_controller C code --------------------
Calling RTW Make Command make_rtw...

### Starting Real-Time Workshop build procedure for model: rtdemo2_1_sm_controller
```

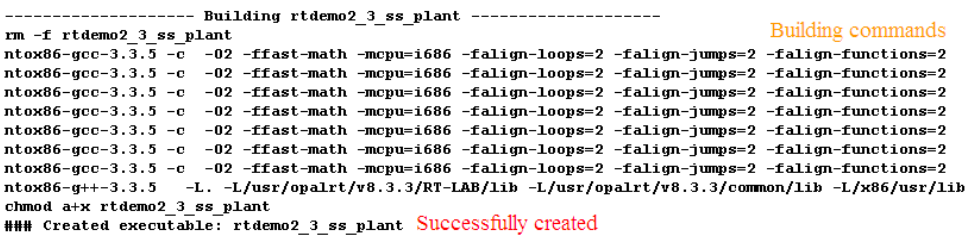
Build, Load & Execute

Build model

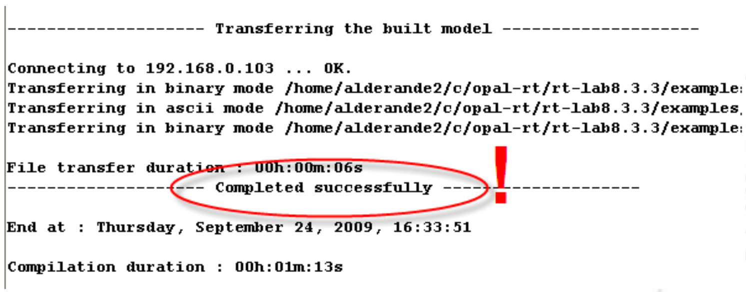
- **Transferring the generated C code**
  - Through an internal RT-LAB process (OpalD)
  - All required files, including files generated by RTW



- **Building the generated C code**
  - The target's compiler builds and links the files to generate a real-time executable



- **Transferring the built model**
  - Executables are transferred back to the host computer
  - Can compile model on Target A and launch it on Target B since all real-time executable files are now located on the host



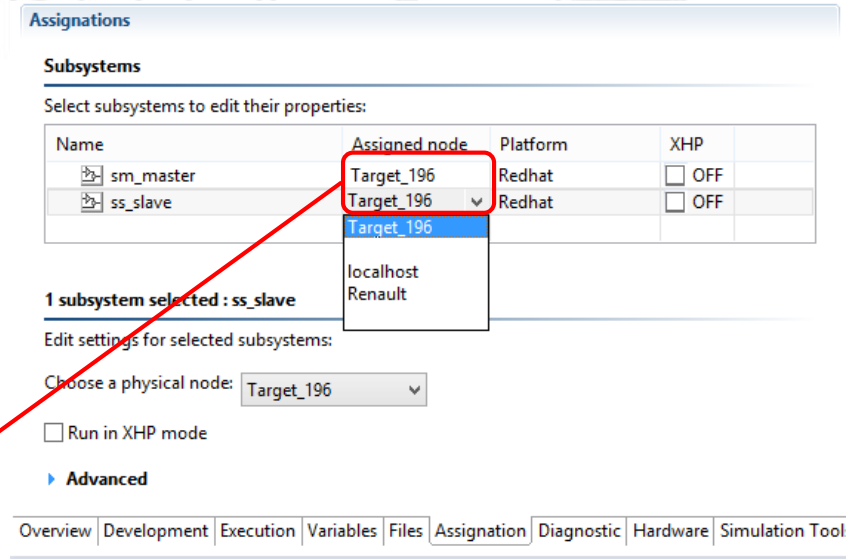
Build, Load & Execute

Load model

- **Many possible configurations**
  - Subsystems can be run on the same target or on different targets
  - Limited by the number of cores

- Example A: 2 subsystems/1 target
  - 2 cores on same target
  - The same target is used twice.

- Example B: 2 subsystems/2 targets
  - 1 core on each target
  - Same model, two targets



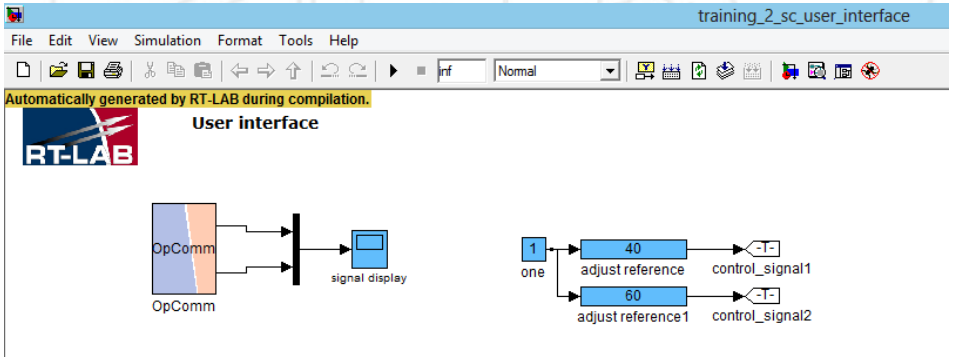
Assign subsystems

- **Select simulation mode**
  - Simulation (offline, as fast as possible)
  - Simulation with low priority (Win32)
  - Software synchronized (real-time)
  - Hardware synchronized (real-time)

Build, Load & Execute

Execute model

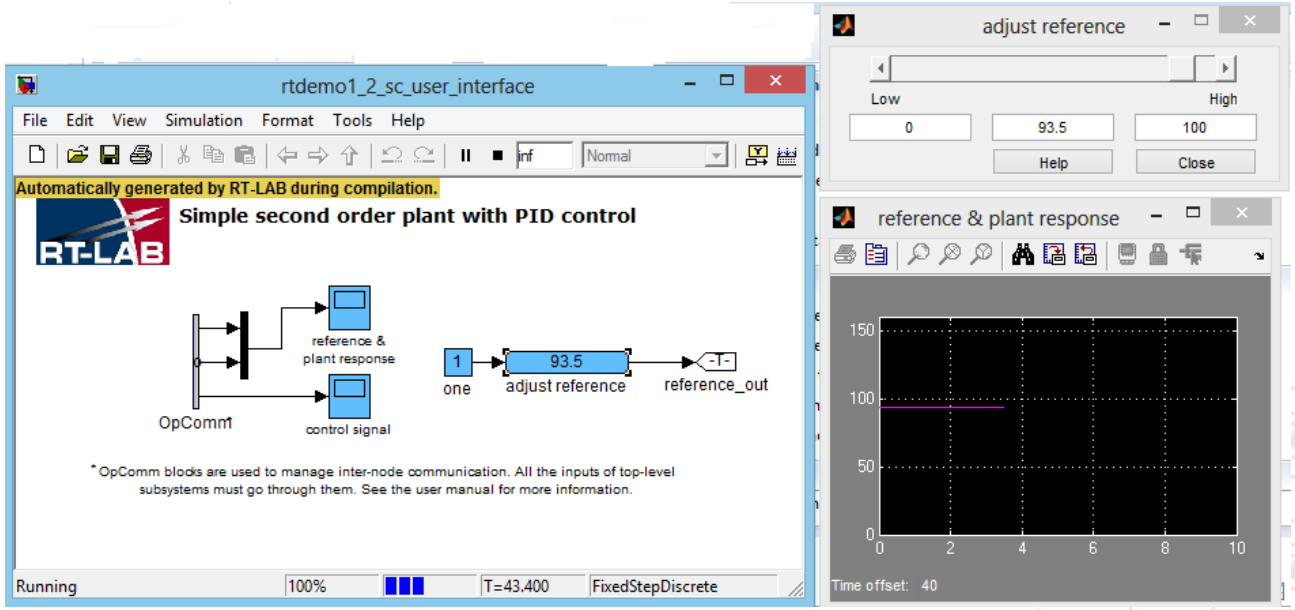
- **Graphical User Interface**
  - Console represents ***SC_subsystem***
  - Modification of parameters on-the-fly is possible



Console

- **Run model**

- **Reset model**



Questions ?