TRNSYS 16

a TRaNsient SYstem Simulation program

Volume 1

Getting Started

Solar Energy Laboratory, Univ. of Wisconsin-Madison
http://sel.me.wisc.edu/trnsys

TRANSSOLAR Energietechnik GmbH
http://www.transsolar.com

CSTB – Centre Scientifique et Technique du Bâtiment
http://software.cstb.fr

TESS – Thermal Energy Systems Specialists
http://www.tess-inc.com

About This Manual

The information presented in this manual is intended to provide a simple guide to get you started using TRNSYS 16. This manual is not intended to provide detailed reference information about the TRNSYS simulation software and its utility programs. More details can be found in other parts of the TRNSYS documentation set. The latest version of this manual is always available for registered users on the TRNSYS website (see here below).

Revision history

- 2004-09 For TRNSYS 16.00.0000
- 2005-02 For TRNSYS 16.00.0037
- 2006-01 For TRNSYS 16.01.0000
- 2006-06 For TRNSYS 16.01.0002
- 2007-03 For TRNSYS 16.01.0003

Where to find more information

Further information about the program and its availability can be obtained from the TRNSYS website or from the TRNSYS coordinator at the Solar Energy Lab:

TRNSYS Coordinator Solar Energy Laboratory, University of Wisconsin-Madison 1500 Engineering Drive, 1303 Engineering Research Building Madison, WI 53706 – U.S.A.	Email: trnsys@engr.wisc.edu Phone: +1 (608) 263 1586 Fax: +1 (608) 262 8464
TRNSYS website: http://sel.me.wisc.edu/trnsys	

Notice

This report was prepared as an account of work partially sponsored by the United States Government. Neither the United States or the United States Department of Energy, nor any of their employees, nor any of their contractors, subcontractors, or employees, including but not limited to the University of Wisconsin Solar Energy Laboratory, makes any warranty, expressed or implied, or assumes any liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

© 2007 by the Solar Energy Laboratory, University of Wisconsin-Madison

The software described in this document is furnished under a license agreement. This manual and the software may be used or copied only under the terms of the license agreement. Except as permitted by any such license, no part of this manual may be copied or reproduced in any form or by any means without prior written consent from the Solar Energy Laboratory, University of Wisconsin-Madison.

TRNSYS Contributors

S.A. Klein	W.A. Beckman	J.W. Mitchell
J.A. Duffie	N.A. Duffie	T.L. Freeman
J.C. Mitchell	J.E. Braun	B.L. Evans
J.P. Kummer	R.E. Urban	A. Fiksel
J.W. Thornton	N.J. Blair	P.M. Williams
D.E. Bradley	T.P. McDowell	M. Kummert
D.A. Arias		

Additional contributors who developed components that have been included in the Standard Library are listed in Volume 5.

Contributors to the building model (Type 56) and its interface (TRNBuild) are listed in Volume 6.

Contributors to the TRNSYS Simulation Studio are listed in Volume 2.

TABLE OF CONTENTS

1. GETTING STARTED

The getting started manual explains what TRNSYS is and what programs make the TRNSYS suite. You will learn how to install TRNSYS, how to open and run the examples, how to create a project in the Simulation Studio and how to use the Multizone Building interface (TRNBuild).

1.1. What is TRNSYS?

TRNSYS is a complete and extensible simulation environment for the transient simulation of systems, including multi-zone buildings. It is used by engineers and researchers around the world to validate new energy concepts, from simple domestic hot water systems to the design and simulation of buildings and their equipment, including control strategies, occupant behavior, alternative energy systems (wind, solar, photovoltaic, hydrogen systems), etc.

One of the key factors in TRNSYS' success over the last 25 years is its open, modular structure. The source code of the kernel as well as the component models is delivered to the end users. This simplifies extending existing models to make them fit the user's specific needs.

The DLL-based architecture allows users and third-party developers to easily add custom component models, using all common programming languages (C, C++, PASCAL, FORTRAN, etc.). In addition, TRNSYS can be easily connected to many other applications, for pre- or post-processing or through interactive calls during the simulation (e.g. Microsoft Excel, Matlab, COMIS, etc.). TRNSYS applications include:

- Solar systems (solar thermal and PV)
- Low energy buildings and HVAC systems with advanced design features (natural ventilation, slab heating/cooling, double façade, etc.)
- Renewable energy systems
- Cogeneration, fuel cells
- Anything that requires dynamic simulation!

Some TRNSYS jargon:

A TRNSYS project is typically setup by connecting components graphically in the **Simulation Studio**. Each **Type** of component is described by a mathematical model in the TRNSYS simulation engine and has a set of matching **Proforma**'s in the Simulation Studio The **proforma has a black-box description of a component: inputs, outputs, parameters,** etc.

TRNSYS components are often referred to as **Types** (e.g. Type 1 is the solar collector). The **Multizone building model** is known as **Type 56**.

The Simulation Studio generates a text input file for the TRNSYS simulation engine. That **input file** is referred to as the **deck file**.

In this manual, %TRNSYS16% refers to the TRNSYS 16 installation directory, i.e. C:\Program Files\Trnsys16 if you chose to keep the default location

1.2. The TRNSYS Suite

TRNSYS consists of a suite of programs: The TRNSYS simulation Studio, the simulation engine (TRNDll.dll) and its executable (TRNExe.exe), the Building input data visual interface (TRNBuild.exe), and the Editor used to create stand-alone redistributable programs known as TRNSED applications (TRNEdit.exe).

1.2.1. The TRNSYS Simulation Studio

The main visual interface is the TRNSYS Simulation Studio (formerly known as IISiBat). From there, you can create projects by drag-and-dropping components to the workspace, connecting them together and setting the global simulation parameters.

The Simulation Studio saves the project information in a Trnsys Project File (*.tpf). When you run a simulation, the Studio also creates a TRNSYS input file (text file that contains all the information on the simulation but no graphical information).

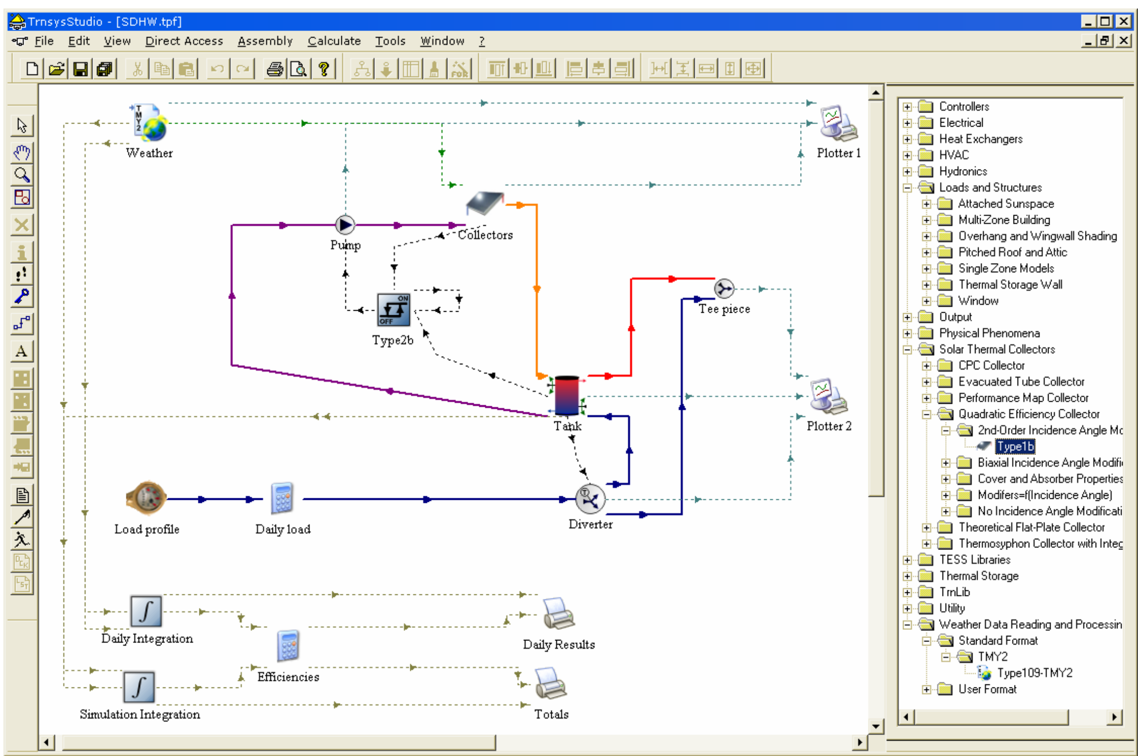

Figure 1–1: The TRNSYS Simulation Studio

The simulation Studio also includes an output manager from where you control which variables are integrated, printed and/or plotted, and a log/error manager that allows you to study in detail what happened during a simulation.

You can also perform many additional tasks from the Simulation Studio: Generate projects using the "New Project Wizard", generate a skeleton for new components using the Fortran Wizard, view and edit the components proformas (a proforma is the input/output/parameters description of a component), view output files, etc.

1.2.2. The TRNSYS Simulation engine

The simulation engine is programmed in Fortran and the source is distributed (see the \SourceCode directory). The engine is compiled into a Windows Dynamic Link Library (DLL), TRNDll. The TRNSYS kernel reads all the information on the simulation (which components are used and how they are connected) in the TRNSYS input file, known as the deck file (*.dck). It also opens additional input files (e.g. weather data) and creates output files.

The simulation engine is called by an executable program, TRNExe.exe, which also implements the online plotter, a very useful tool that allows you to view dozens of output variables during a simulation.

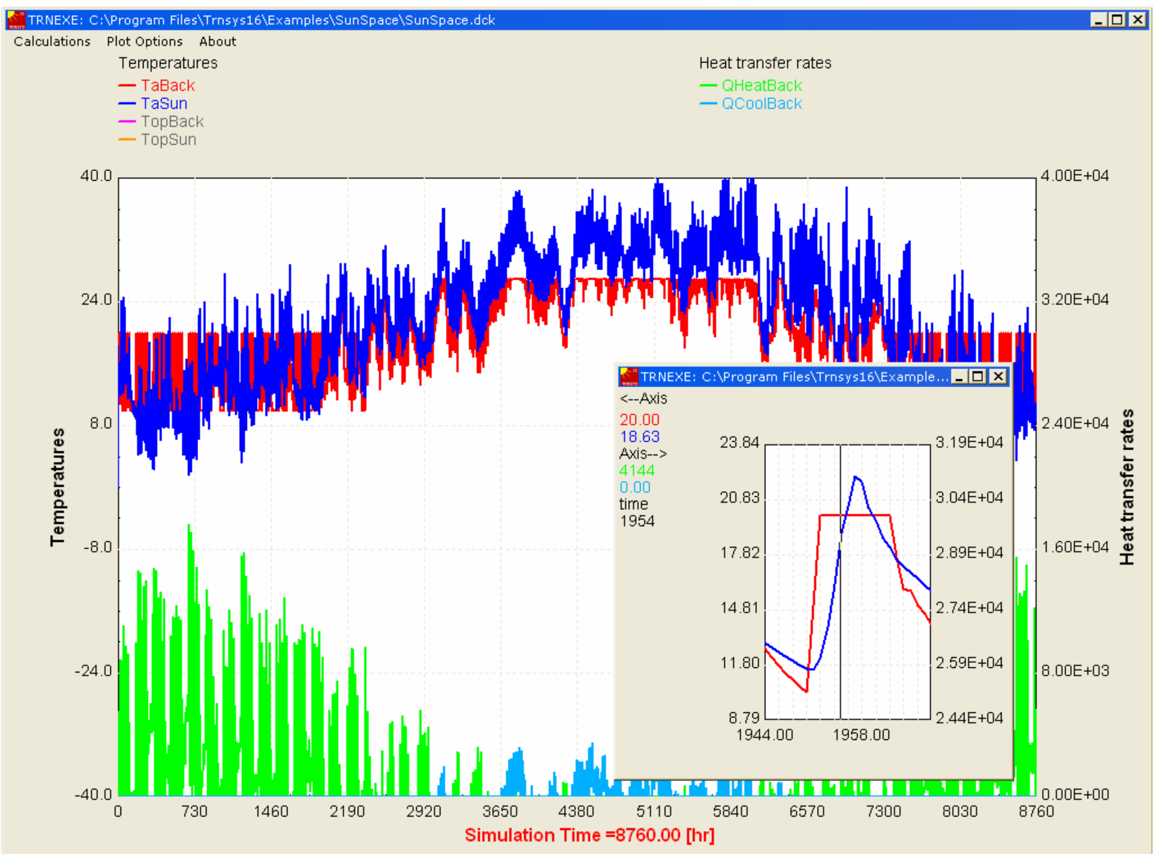

Figure 1–2: The online plotter in TRNExe

The online plotter provides some advanced features such as zooming and display of numerical values of the variables at any time step, as shown in the zoom part of Figure 1–2.

1.2.3. The Building visual interface

TRNBuild (formerly known as Prebid) is the tool used to enter input data for multizone buildings. It allows you to specify all the building structure details, as well as everything that is needed to simulate the thermal behavior of the building, such as windows optical properties, heating and cooling schedules, etc.

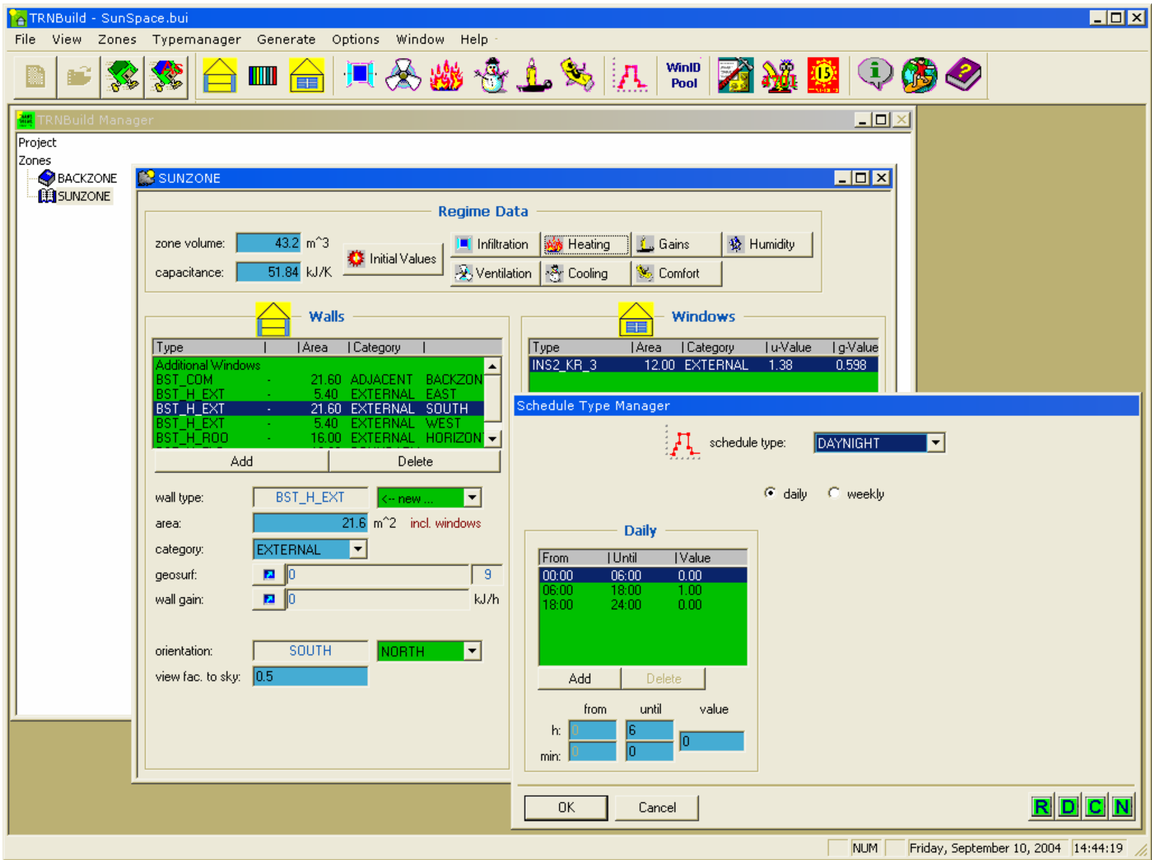

Figure 1–3: TRNBuild

TRNBuild creates a building description file (*.bui) that includes all the information required to simulate the building. Note that this was not the case in TRNSYS 15 where the window library (w4-lib.dat) was always required to run a simulation.

1.2.4. TRNEdit and TRNSED applications

TRNEdit is an specialized editor that can be used to create or modify TRNSYS input files (decks). This is not recommended in general and only advanced users should attempt to modify deck files by hand. Most users should rely on the Simulation Studio to generate and modify deck files.

On the other hand, TRNEdit can be used to create redistributable applications (known as TRNSED applications). Those executables can be freely distributed to end-users who do not have a TRNSYS license in order to offer them a simplified simulation tool. The distributable includes a dedicated visual interface designed by adding special commands to the TRNSYS input file. Advanced features, such as multiple windows (tabs) and clickable pictures, have been added in TRNSYS 16.

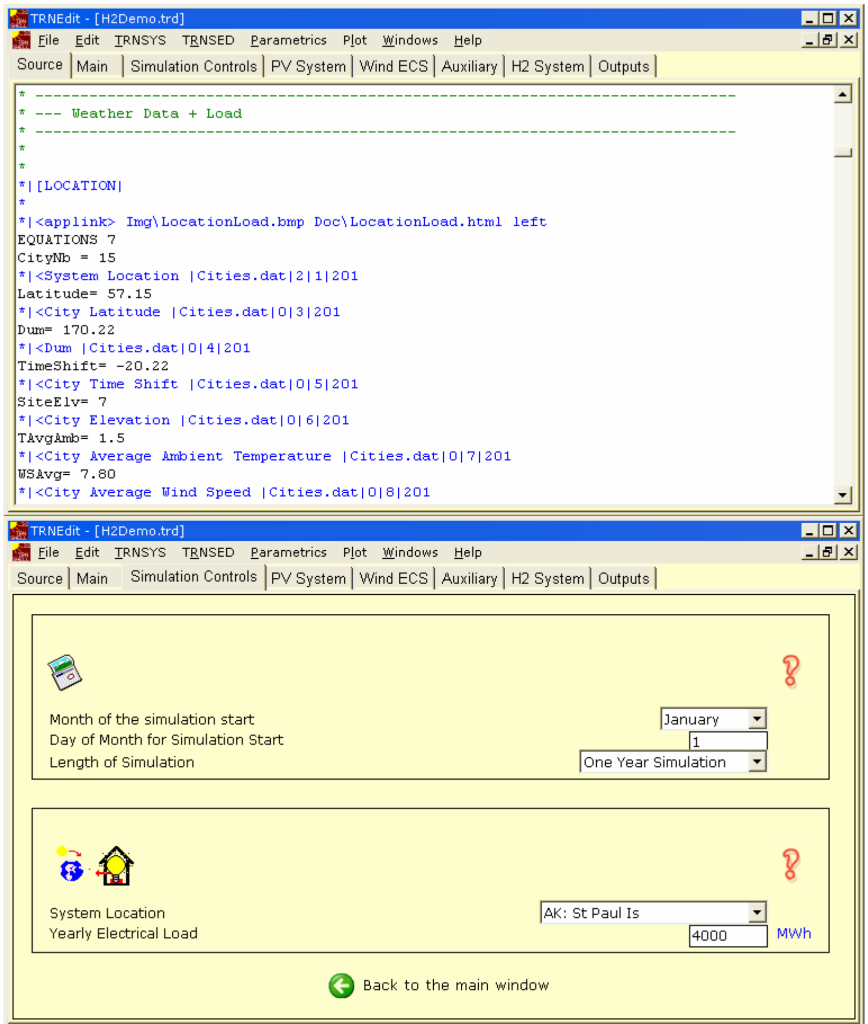

Figure 1–4: TRNEdit – Tabbed view to design TRNSED applications

For more information on TRNSED applications licensing, please check the license.txt file in your installation.

1.2.5. TRNSYS add-ons

TRNSYS offers a broad variety of standard components, and many additional libraries are available to expand its capabilities:

- TRNLIB: sel.me.wisc.edu/trnsys/trnlib (free component library)
- TRANSSOLAR libraries: www.transsolar.com
- TESS libraries: www.tess-inc.com

1.3. Installation

1.3.1. System requirements

1.3.1.1. Software

OPERATING SYSTEM

TRNSYS requires Windows 95/98, NT 4.0, 2000, ME or Windows XP. Windows 2000 or XP are recommended due to known issues in resource management in previous Windows versions (those issues are not related to TRNSYS nor its utility programs).

COMPILERS

Thanks to the drop-in DLL technology, no compiler is required to add components to TRNSYS if you obtain them as a precompiled DLL.

If you wish to debug TRNSYS or to add components for which you received the Fortran code, you will need a Fortran compiler (Please see the Programmer's Guide for supported compilers). If you wish to create your own components, you will need a compiler capable of creating a DLL. This includes any recent Windows-compatible compiler for C++, Delphi, Fortran or other languages.

OTHER

Viewing the documentation requires the free Acrobat reader (Version 6.0 or higher is recommended to benefit from advanced searching capabilities).

1.3.1.2. Hardware

TRNSYS requirements will mostly depend on the simulations you want to run. The specifications here below list the basic requirements and the recommended configuration. The latter should allow you to run yearly simulations with a 0.1 sec time step using the full capabilities of the online plotter, while the basic configuration will allow you to run more typical simulations.

MINIMUM CONFIGURATION

- Pentium III processor or equivalent
- 128 MB of RAM
- 200 MB of available disk space (up to 2 GB if you install all the optional weather data files)

RECOMMENDED CONFIGURATION

- Pentium IV processor 2.0 GHz or equivalent
- 512 MB of RAM
- 1 GB of available disk space (up to 3 GB if you install all the optional weather data files)

1.3.2. Installing TRNSYS 16

Run the Setup program (trnsys16-setup-16-xx-xxxx.exe), where xx.xxxx is the exact release number. The Setup program will guide you through a series of dialog boxes with simple options. You will be prompted to accept the license agreement to continue the installation.

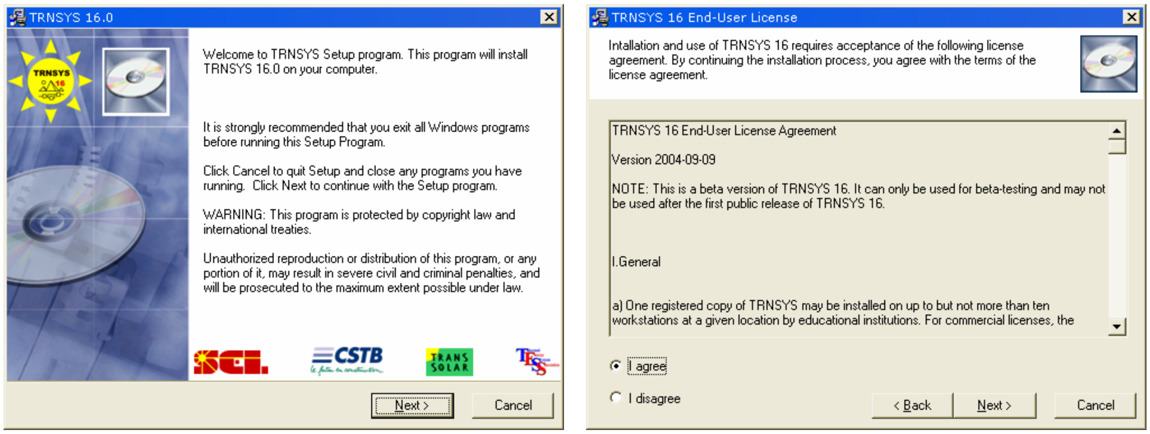

Figure 1–5: Installing TRNSYS 16 - Part 1

You can navigate between the dialog boxes using the Back and Next Buttons. You will go through a few basic options.

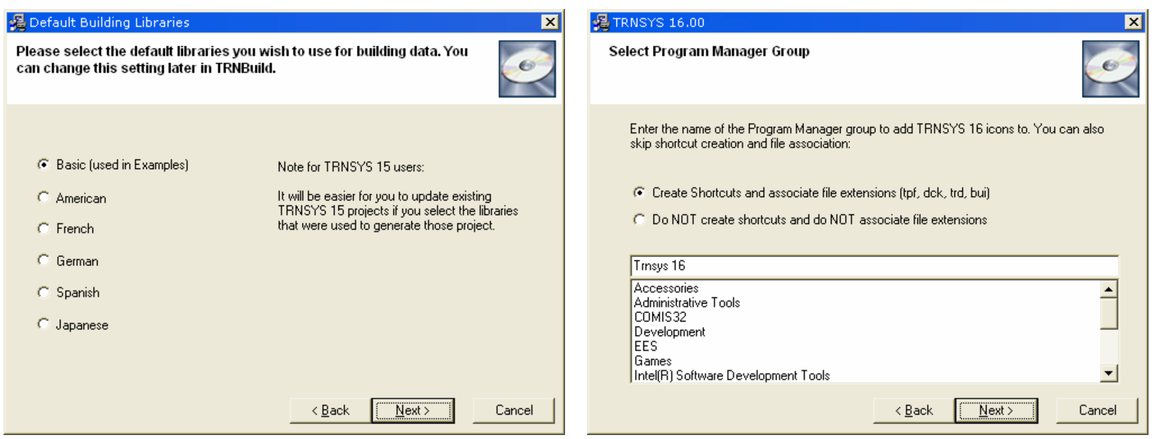

Figure 1–6: Installing TRNSYS 16 - Part 2

You will be asked to choose a default set of libraries for building data (materials and glazing). If you have many TRNSYS 15 projects that you need to update, choose the libraries that you were using in TRNSYS 15. If your country is listed, use those libraries (they were created by local distributors and are often in languages other than English). If you do not know what to do, keep the default choice (Basic). Note that you can always change the default libraries later in TRNBuild. You will then be asked if you want to create shortcuts and associate file extensions with TRNSYS. You should select that option unless you have already installed TRNSYS 16 and you have customized your shortcuts or associations.

When you have selected a few options such as the destination directory, Setup will ask your confirmation to start the installation.

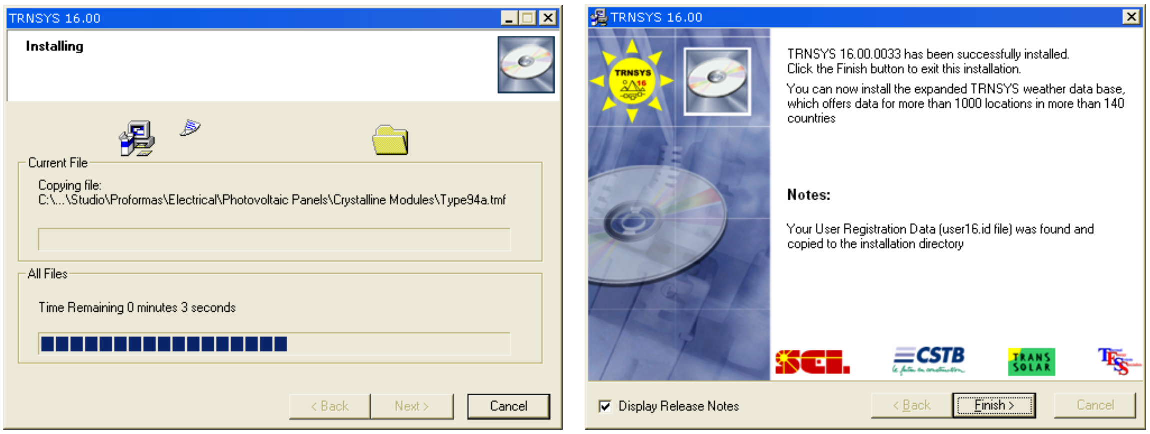

Figure 1–7: Installing TRNSYS 16 – Part 3

When the installation is complete, the Setup program will attempt to copy your user registration data (the "user16.id" file) to the destination folder. If you install from a CD provided by your distributor, Setup should be able to locate the file. If it is not the case (e.g. if you are installing a downloaded version), you should copy the "user16.id" file provided by you distributor according to the message displayed on the last screen of the Setup program.

When you click "Finish", Setup will exit and launch your default browser to display the release notes of the version you have just installed. You can skip this by unchecking the "Display Release Notes" checkbox on the last screen of the Setup program (this is not recommended as the Release Notes have important information on the program you have just installed).

1.3.3. Installing the expanded weather data

After installing TRNSYS 16, the setup program will try to copy your user registration data to the destination directory. The registration information is contained in a small text file, user16.id. If you install from a CD obtained from your distributor, Setup should find the user16.id file and install it. If you downloaded TRNSYS from the web, Setup will inform you in case it can't find the user16.id file. In that case you just need to copy that file to the installation directory (C:\Program Files\Trnsys16 by default). Please contact your distributor if you cannot locate your registration information.

The TRNSYS Setup installs a few weather data files that will show you an example of the available data sources and data formats. However, TRNSYS 16 comes with a comprehensive set of weather data files including more than 1000 locations in more than 140 countries. Please refer to Volume 6 of the documentation (Weather Data) for more information on the available data sources and locations.

If you wish to proceed with the installation of the expanded weather data base, just run the weather data Setup program, trnsys16-weather-16-xx.exe (where xx is the release number). The Setup program is very similar to the TRNSYS 16 installation program and will lead you through a few steps with dialog boxes.

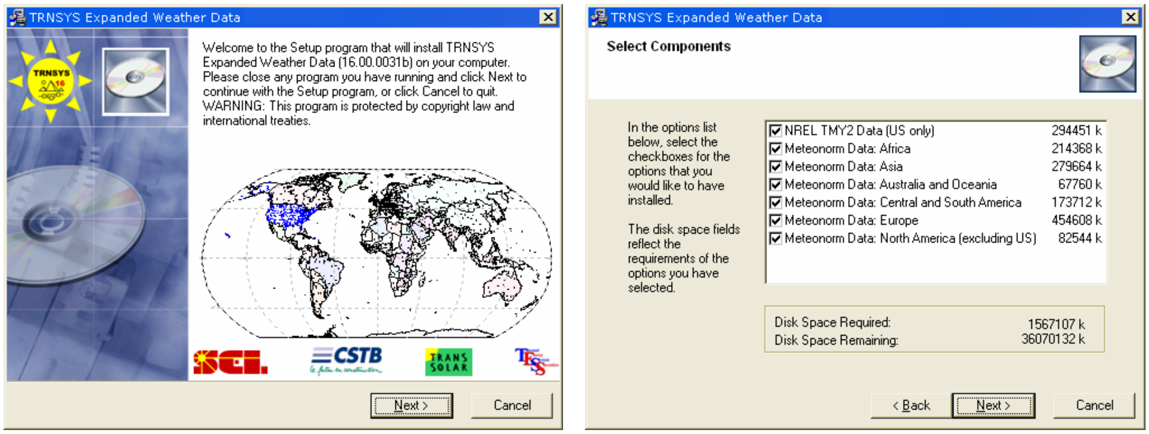

Figure 1–8: Installing the expanded weather data

You can choose which datasets to install or not (the available packages may not match the screenshot here below depending on your version). Please note that installing all the data files will require more than 1.5 GB of available disk space on your computer.

When the weather data installation is completed, you are ready to use TRNSYS.

1.4. Using TRNSYS examples

This section explains how you can quickly get started using TRNSYS by opening and running the examples provided with the distribution. You can then start creating your own projects by making changes to those examples.

1.4.1. Opening and running a simple example

Launch the simulation Studio by using the created shortcut or by browsing to %TRNSYS16%\Studio\Exe and launching the Program called Studio.exe

Go to File/Open and select %TRNSYS16%\Examples\Begin\Begin.tpf. A TRNSYS project consists of components (e.g. a solar collector, a data reader, a printer) linked together.

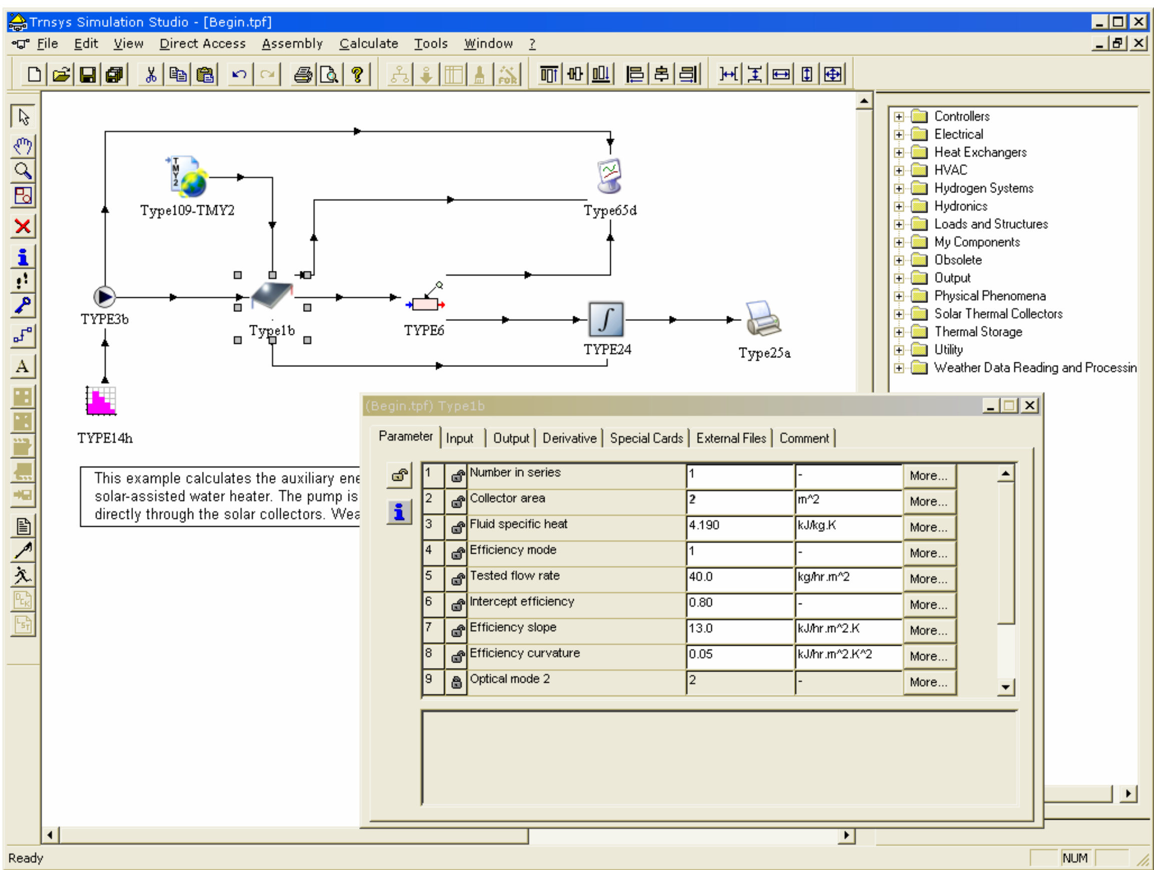

Figure 1–9: The "Begin" example

Note: Each component is assigned two numbers: a Type and Unit numbers. If you press the key F2 you will see these two values for all of the components present in the simulation.

1.4.1.1. *Component configuration*

You can check a component's configuration by double-clicking its icon. This will open a window with multiple tabs. When you open the window, the foremost tab shows a list of parameters and their value (the solar collector parameters are shown in Figure 1–9). You can see additional information about the parameters by clicking the "More" button.

You can explore the different tabs to view the component's inputs, outputs and derivatives (if any – derivatives are capacitive variables of the component, e.g. nodes representing a given amount of water in a storage tank).

> Note: The values and units displayed in the input tab give the initial values for the corresponding inputs. They are overridden during the simulation if those inputs are connected to other components

1.4.1.2. *Connections*

If you double-click on a link between two components, you will open a new window that lists all input-output connections inside that link. Figure 1–10 shows the link between Type 109 (weather data reading and processing) and Type 1b (solar collector)

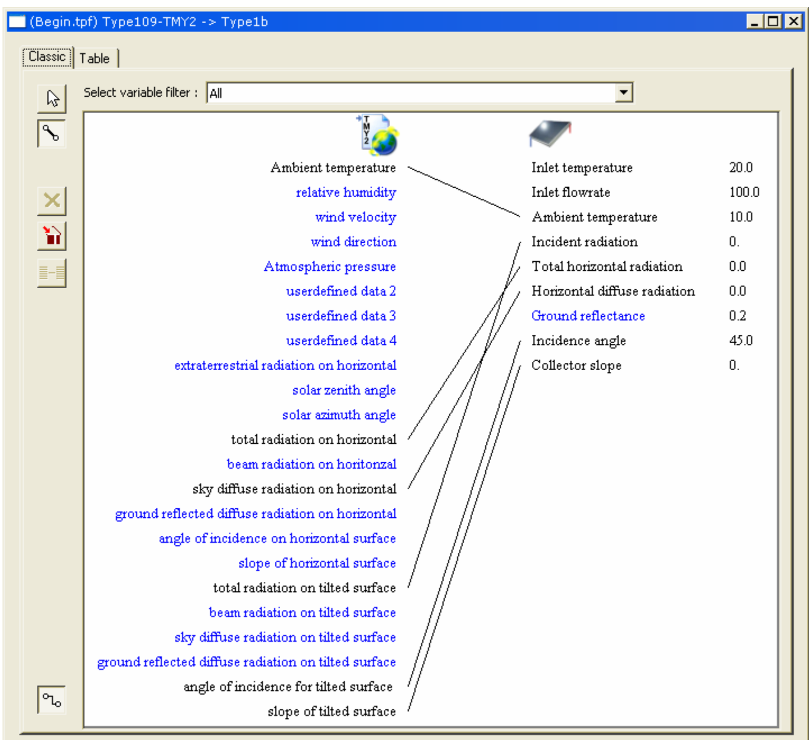

Figure 1–10: Example of connection window

If there are many available inputs and outputs, the lists will be longer than what can be displayed. You can resize the window and/or use the scrollbars.

> Note: The Simulation Studio can be set to auto-scroll in the connection window (as well as in the project window). If you want to enable/disable that mode, go to File/Settings/Project and check/uncheck the "auto-scroll" boxes.

it may be easier to make new connections after aligning a given output with the input to which it is connected. To do this, click on the bottommost icon on the left (see Figure 1–11 for an example).

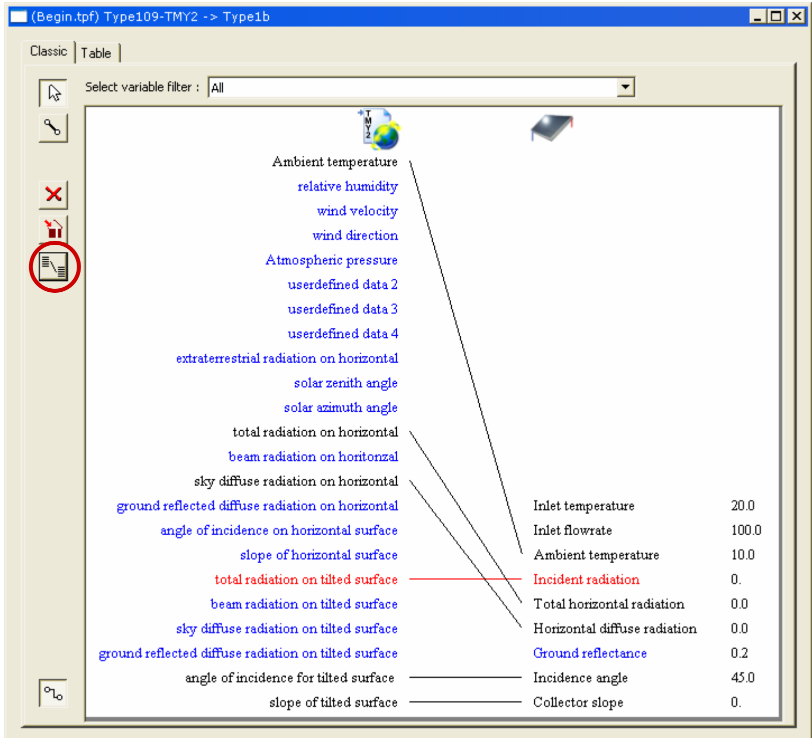

Figure 1–11: Aligned connection window

You can also decide to show only the inputs and outputs that have given dimensions. Figure 1–12 shows the results when "angle" is selected. Note that inputs/outputs with "any" or "unknown" dimensions will always be displayed.

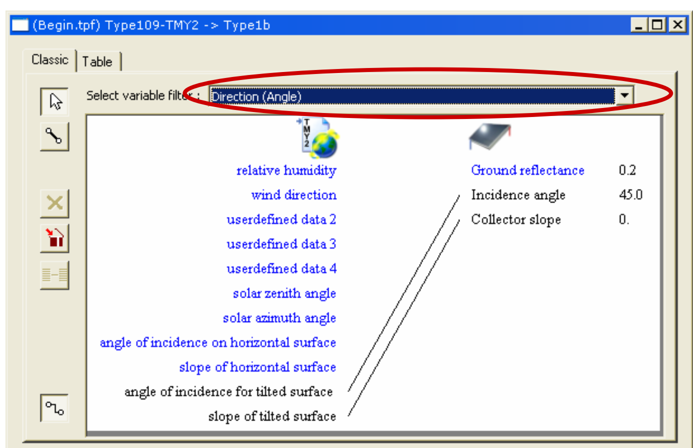

Figure 1–12: Filtered connection window

Alternatively, you can use the table view to display and manage connections: Just switch from the "Classic" tab to the "Table" tab. The result is shown in Figure 1–13.

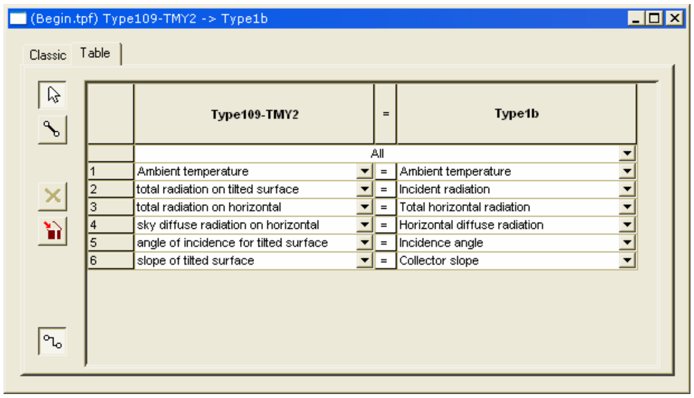

Figure 1–13: Aligned connection window

1.4.1.3. Running the simulation and viewing the results

You can run a simulation by pressing the "F8" key, which is the shortcut for "Calculate/Run Simulation".

THE ONLINE PLOTTER

If at least one "online plotter" component is present in the simulation, an online plot will be displayed during the simulation. The online plotter offers several features that will help you analyze the simulation results while it is running and after it is done.

You can interrupt / resume the simulation while it is running by right-clicking anywhere in the plot, by using the "F7" and "F8" keys, or using the "Calculation/Stop" and "Calculation/Resume" menu entries. The "Pause at…" command is also very useful when you want to diagnose some problems occurring at a given time in a simulation.

When the simulation is stopped, you can use the "Plot options" menu to change the plot background from black to white, or increase the line thickness. You can also change the left and right Y-axis limits by clicking on the axes themselves, which will display a dialog box (see Figure 1–14). Please note that changes to those limits will be lost if you re-run the simulation. You should change the online plotter parameters in the simulation itself (double-click on the online plotter icon) if you want changes to be permanent.

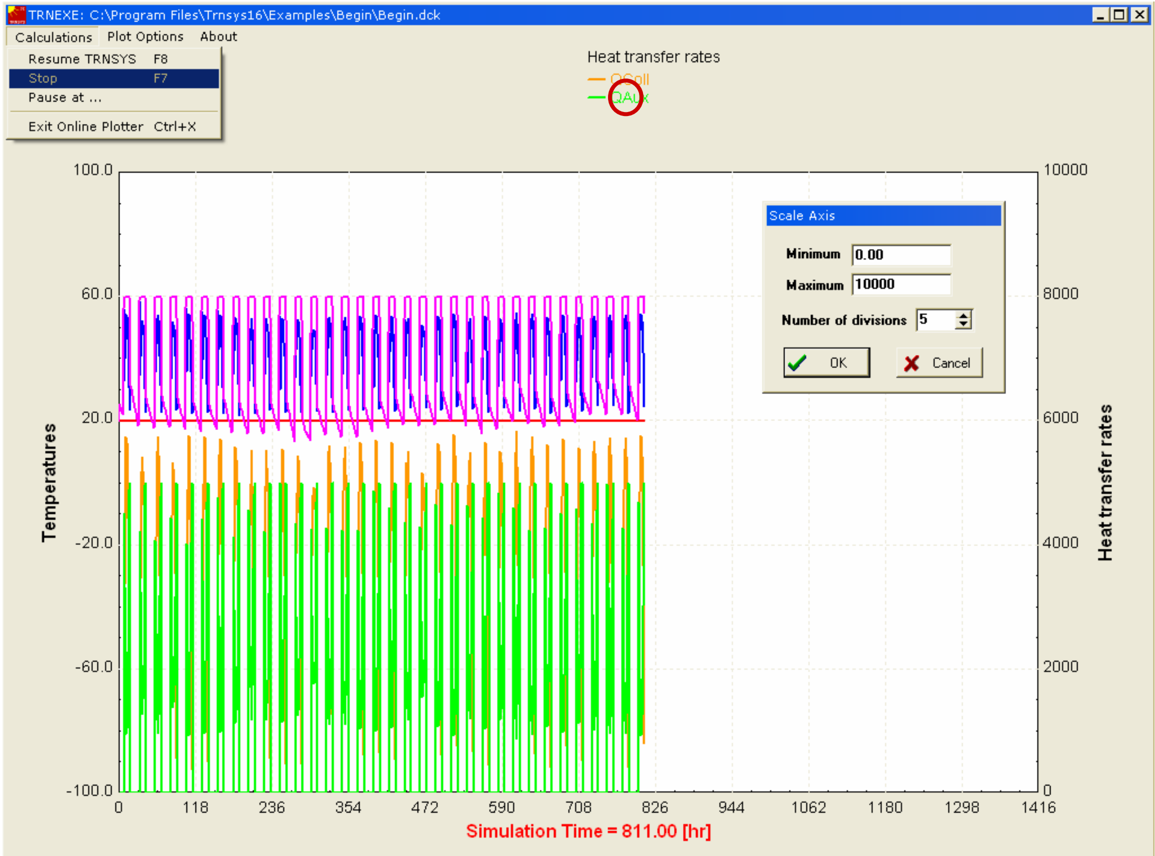

Figure 1–14: Online plotter in a paused simulation with Y-axis control box

You can hide or show any variable in the plot by clicking on its name in the legend fields. For example clicking in the red circle in Figure 1–14 would hide/show the QAux plot.

ANALYZING THE SIMULATION: ZOOMING AND DISPLAYING NUMERICAL VALUES

You can zoom on part of the plot to have a more detailed view of a shorter time interval. Just click on the upper-left corner of the area you want to zoom in and drag the mouse pointer to the lower-right corner, then release the mouse button. In the zoom window, you can adjust the Y-axis limits **but also the X-axis (time) limits** by clicking on the axes. This is very useful when you want to study such a short period of time that it is hard to zoom on that period right away.

You can display the numerical value of any variable at any point in time in both the "normal" and the "zoom" windows. Press the SHIFT key and mover the mouse over the graph. The variable labels will be replaced with their value (and "time" will be replaced with the simulation time). This is shown in Figure 1–15 for the zoom window.

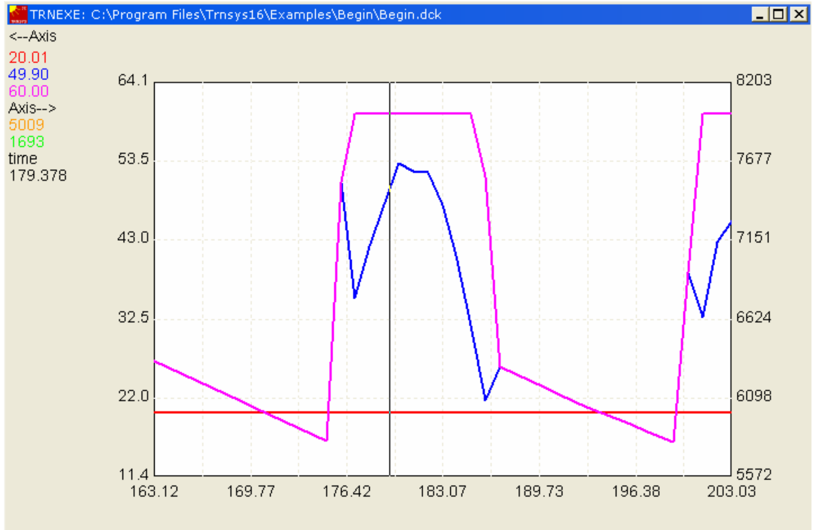

Figure 1–15: Online Plotter: Displaying numerical values

Note: By pressing SHIFT and moving the mouse over the plot, you will display the values plotted by the online plotter, which are interpolated between TRNSYS time steps. If you want to see only the actual simulation time steps, pres CTRL-SHIFT when moving the mouse. This can be useful to study control signal switching from 0 to 1, for example, since the online plotter will draw a continuous line between those 2 states and it will show interpolated values that do not correspond to any simulated values.

CLOSING THE ONLINE PLOTTER AND ANALYZING PRINTED RESULTS

At the end of the simulation, you will be asked if you wish to exit the online plotter. If you click "No", you will be able to use the online plotter commands described here above. If you click "Yes", you will come back to the Simulation Studio, from where you can view the printed results.

You can open external files (input or output files) by using the "Calculate/Open/External Files" menu or by double-clicking on the component that uses a file, switching to the "External Files" tab and using the "Edit" button (Figure 1–16). Both actions will open the file using the editor set in "File/Settings/Directories/Text Editor" (Notepad by default).

Note: In a file name, "***" means that TRNSYS will use the input file (.dck) filename to assign a name at the file at runtime. Example: if your project's input file is called "MyProject.dck" and you type in "***.dat" as the output file name, TRNSYS will create a file called "MyProject.dat".

Warning: the input (deck) file name is not always the same as the TRNSYS Studio project's name. The deck filename is set in the project's Control Cards, which can be accessed by "Assembly/Control Cards" or by the appropriate toolbar button

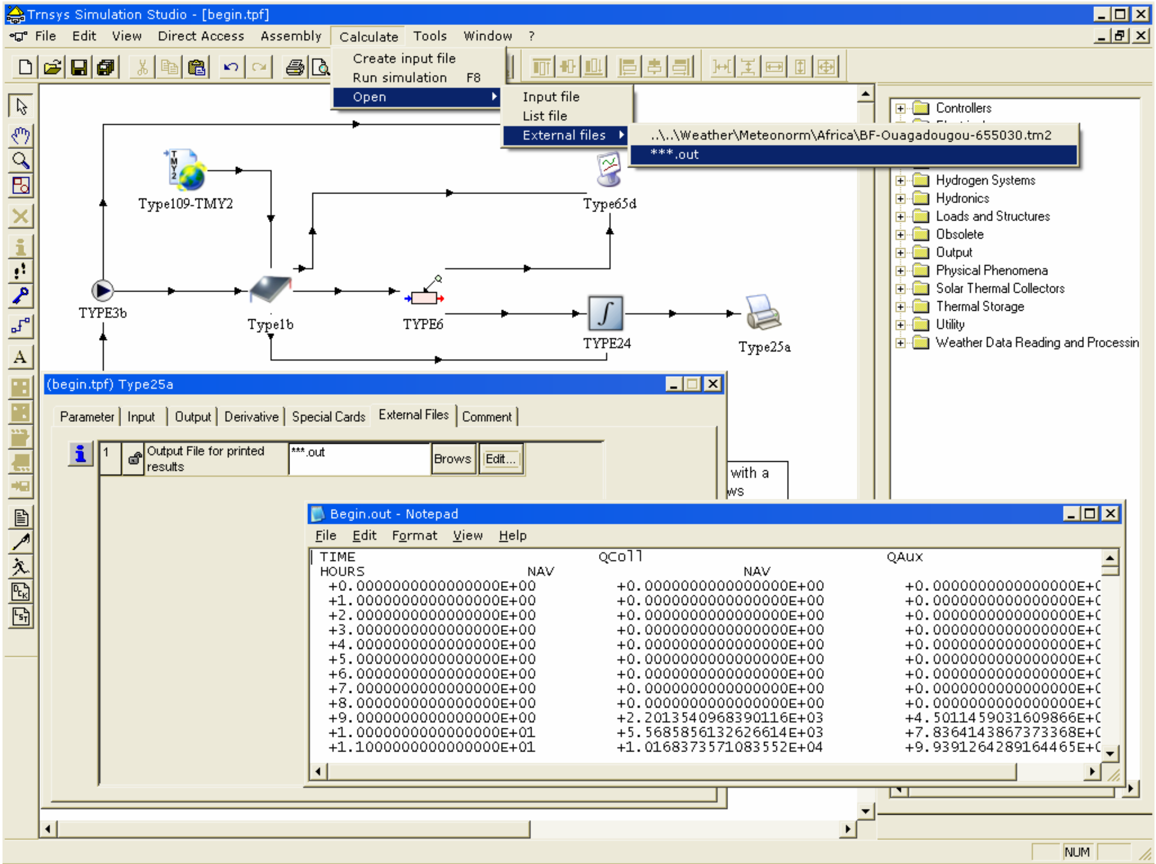

Figure 1–16: Opening external files

The Standard TRNSYS components always create text files, but you can use any file extension in a project. In particular, some users find it convenient to use the file extension registered with their preferred spreadsheet program, e.g. ".xls" for Microsoft Excel. This allows opening those files in the spreadsheet program by double-clicking their name in Windows Explorer. Please note, however, that the created files only have plain text information. Special features, like colors, cannot be created with the standard components.

TROUBLESHOOTING A SIMULATION (THE ERROR MANAGER)

During a simulation, TRNSYS writes messages to a special file called the Log file. That file has the same name as the input file (deck) with a ".log" extension. Another file, the Listing file, is also created (the listing file also has all messages but in addition it repeats the input file and has some additional printed outputs like the results of a "Trace" command, which prints the inputs and outputs of a component at each iteration).

The Simulation Studio provides access to the Log and Listing file through the Error Manager, which can be accessed by clicking on the LST button. Figure 1–17 shows an example of error message when an equation refers to a non-existent variable. The TRNSYS simulation ends with a "TRNSYS Errors" dialog box. You can then return to the Simulation Studio by clicking OK, and you can launch the Error Manager to analyze notices, warnings and error messages that were generated during the simulation.

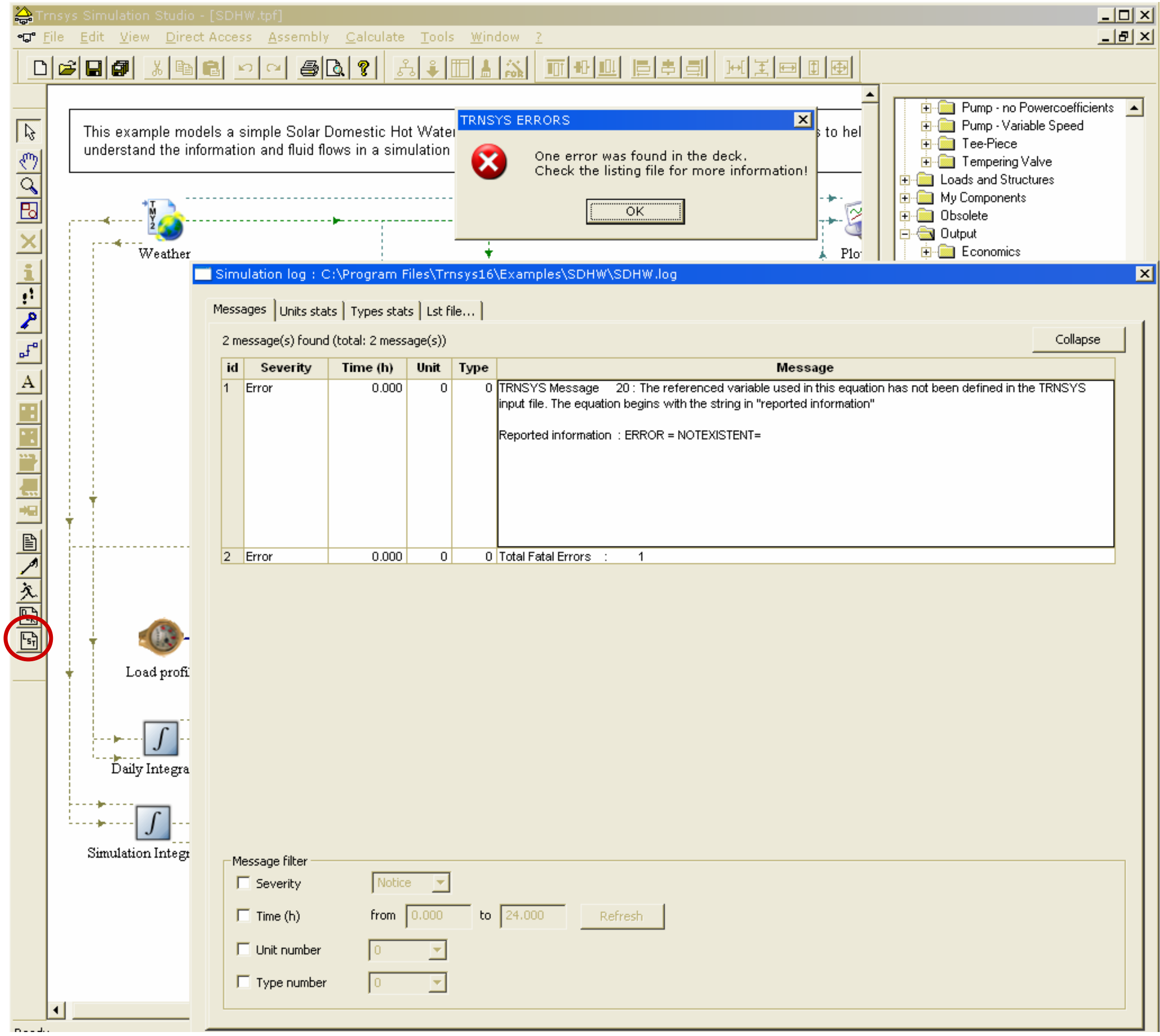

Figure 1–17: The Error Manager

The "Units stats" and "Types stats" of the Error Manager present additional information on the calculation time spent in each component and on the number of times each component was called. Finally, clicking on the "Lst File" tab will open the listing file in a text editor.

1.4.2. Opening and running an example with the Multizone building model (Type 56)

The "Sunspace" example is a simple example inspired by BESTEST Case 960. BESTEST (Building Energy Simulation programs TEST) is the methodology developed in the framework of the IEA to test and diagnose the simulation capabilities of the exterior envelope portions of building energy simulation programs.

1.4.2.1. *Opening and running the example*

In the simulation Studio, open %TRNSYS16\Examples\SunSpace\SunSpace.tpf. You can explore the connections and the components configuration as explained here above.

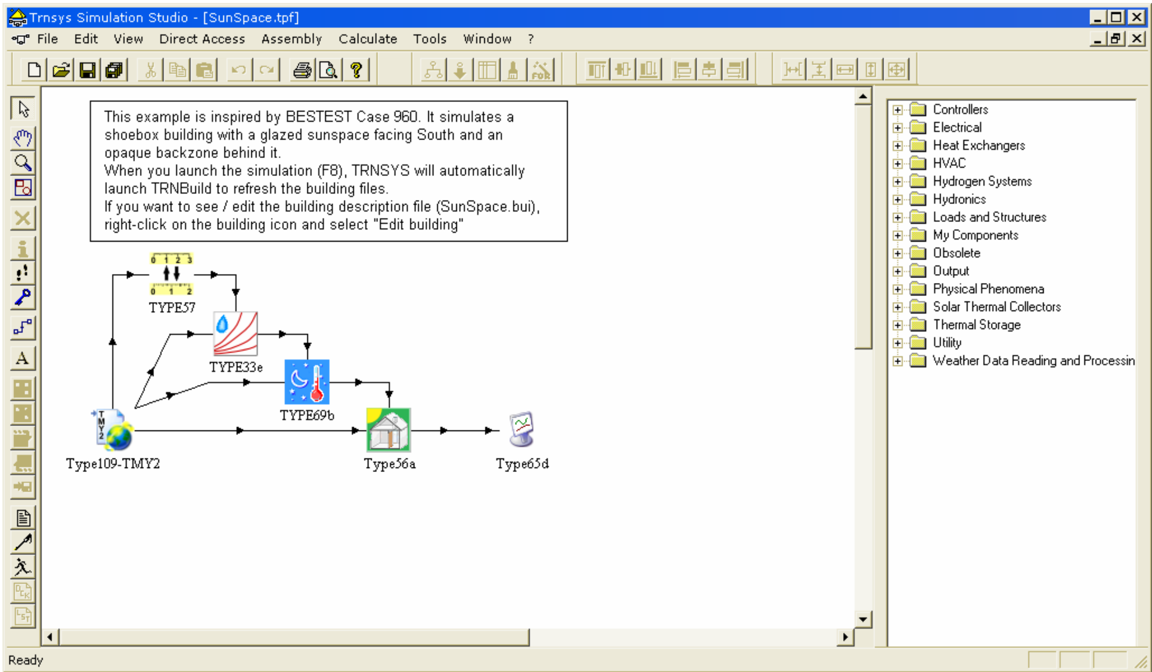

Figure 1–18: The SunSpace example

When you run the example (F8), TRNSYS launches TRNBuild in order to process the building input data. This ensures that the data used in TRNSYS matches the latest version of the .bui file and that Type 56 will find all intermediate files it uses for the simulation (.bld, .trn and .inf). After the automatic call to TRNBuild, the online plotter is displayed and the simulation runs.

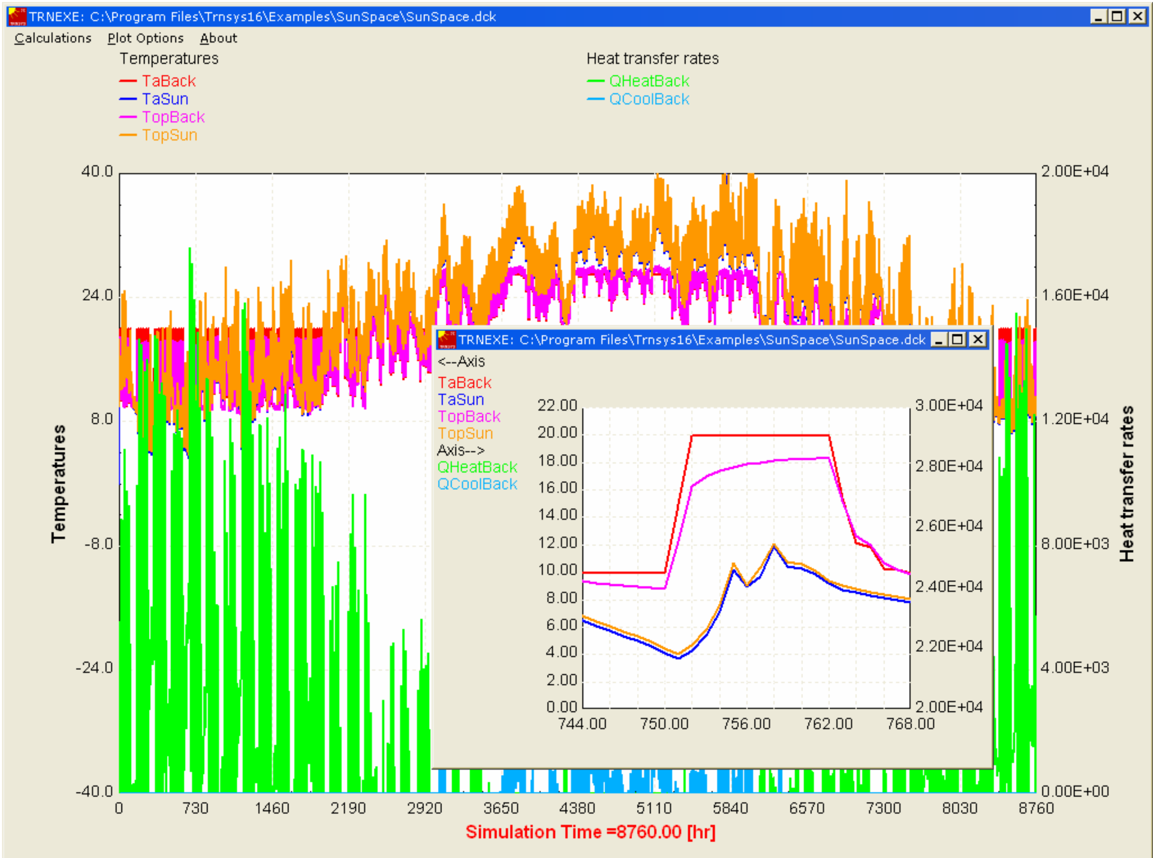

Figure 1–19: SunSpace example: online plotter with zoom on air and operative temperatures

Figure 1–19 shows the online plotter with a zoom window displaying the air and operative temperatures on February 1st.

1.4.2.2. Editing the building description

A building model involves too many parameters to use a standard proforma in the Simulation Studio. The building model is described in a special file, the .bui file. You can edit the building description by right-clicking on the building icon and selecting "Edit Building", as shown in Figure 1–20. This will launch TRNBuild and open the corresponding .bui file.

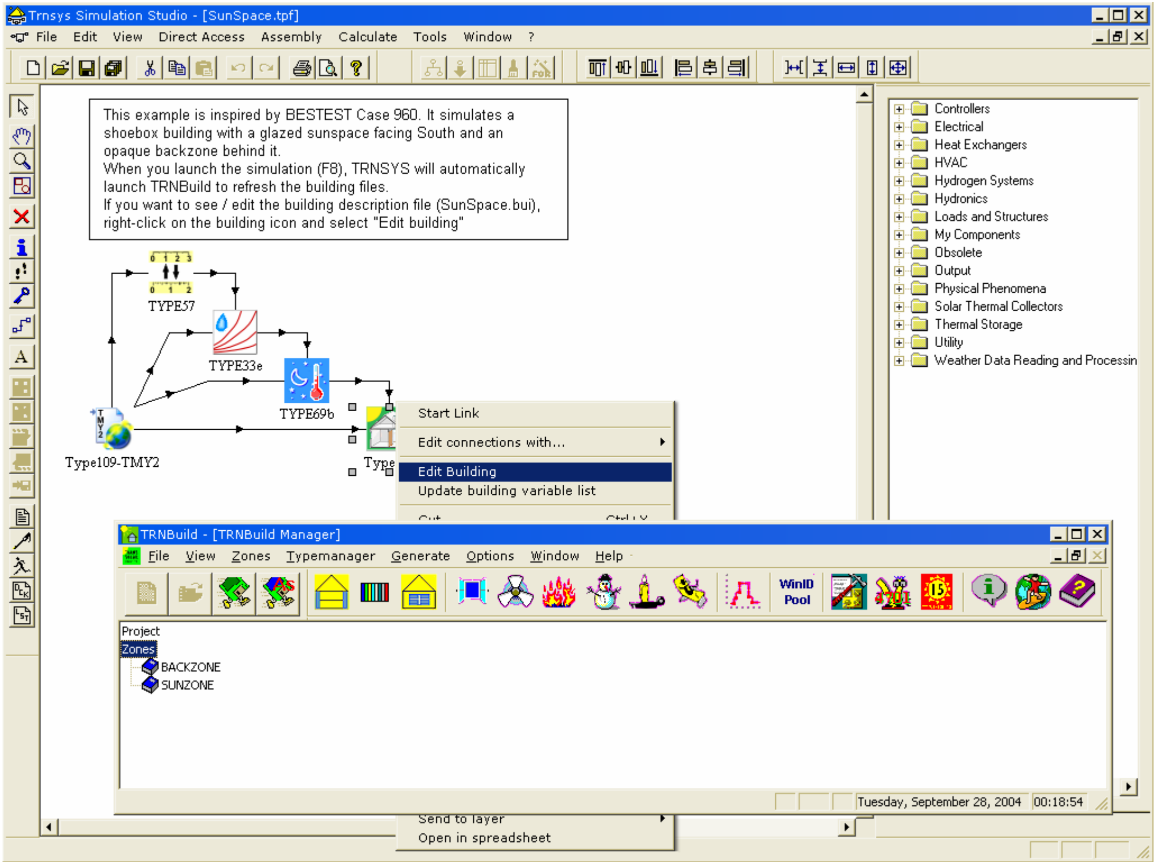

Figure 1–20: SunSpace example: Launching the Building Editor, TRNBuild

You can explore the building parameters from the top level (thermal zones) to the bottom level (thermal properties of one layer in a massive wall). You can, for example, change the window area in the zone "SunZone" from 12 to 1, as illustrated in Figure 1–21. To do this, click on the name of the zone (SUNZONE) in the TRNBuild Manager, then in the zone window select the 3rd wall (BST_H_EXT wall Type with orientation SOUTH). The properties of windows pertaining to that wall will be displayed in the right column.

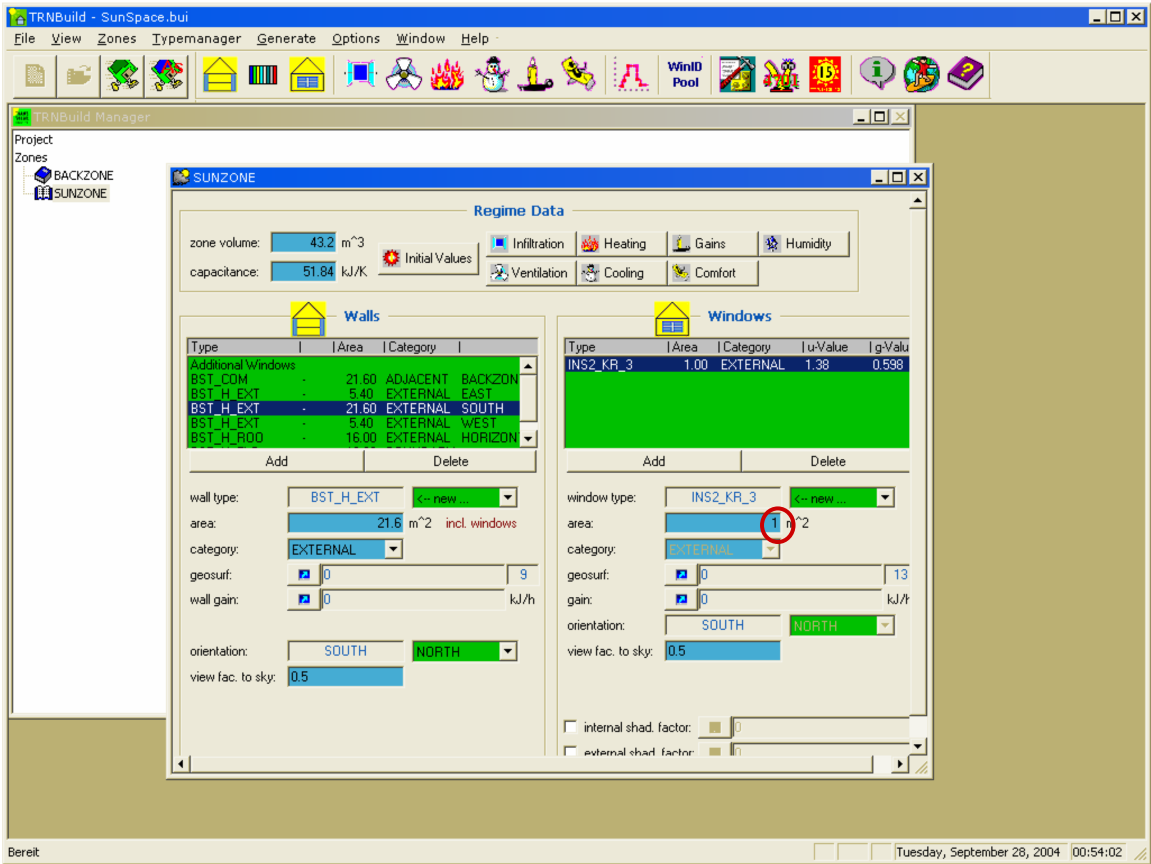

Figure 1–21: SunSpace example: changing the window area

After changing the area, you will notice a significant decrease in summer temperatures compared to Figure 1–19.

1.5. Creating a TRNSYS project

In this section, we will build the "Begin" example from scratch, starting from a physical description of the system that has to be modeled.

1.5.1. System description

The simulated system is represented in Figure 1–22. It is a simple solar thermal application where solar collectors are used to preheat water in an industrial process.

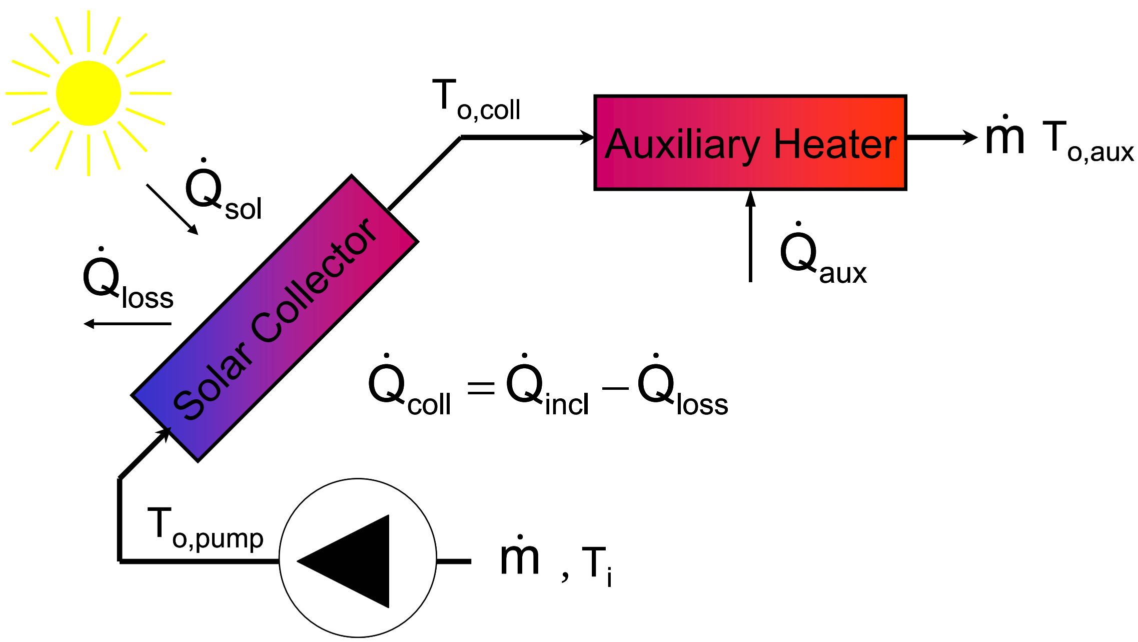

Figure 1–22: A simple solar pre-heating application

The water is pumped from 8 am to 6 pm every day with a pump that has a maximum flow rate of 50 liters per hour. The supply water enters the pump at a constant temperature of 20°C. The pump electrical power is 16.7 W and it is assumed that 5% of the electrical power, converted to heat, is transferred to the fluid.

The solar array consists of two 1-m² collectors connected in parallel. 1 m² is the aperture area (glazing area for a flat-plate collector), which is consistent with the efficiency parameters given here below. Both collectors are identical. Their performance has been tested with a flow rate of 40 l/h-m² and the efficiency parameters were found to be η_0 = 0.8, a_1 = 3.61 W m^{-2} K^{-1}, a_2 = 0.014 W m^{-2} K^{-2}, with:

$$\eta = \eta_0 - a_1 \frac{T_i - T_a}{G} - a_2 \frac{(T_i - T_a)^2}{G} \qquad \left(\eta = \frac{\dot{Q}}{A_a G} \right),$$

where

η	Collector efficiency	[-]
η_0	Intercept (maximum) efficiency	[-]
a_1	First order loss coefficient	[W m^{-2} K^{-1}]
a_2	Second order loss coefficient	[W m^{-2} K^{-2}]
G	Solar radiation in the collector plane	[W m^{-2}]
T_i	Fluid inlet temperature	[°C]
T_a	Ambient air temperature	[°C]
$\dot{Q}$	Useful Heat Transfer Rate of the collector	[W]
A_a	Aperture area of the collector	[m²]

Furthermore, the zero and first order incident angle modifiers according to ASHRAE testing method were found to be b0 = 0.2 and b1 = 0.0. The solar collectors have a slope of 45° and face due South (note: this is not intended to provide an example of optimal solar collector orientation for the given location).

> In Europe, solar collector test reports often give the efficiency in a slightly different way from here above: The average (or mean plate) temperature is used instead of the inlet temperature. Make sure to take this into account in the efficiency mode (Parameter 4) of Type 1 if this is the case (that parameter should then be set to 2)

The auxiliary heater has a maximum power of 1400 W and losses should not be taken into account. The set point is 60°C.

The system is located in Ouagadougou, Burkina Faso. Expected results from the simulation are a hourly plot of system temperatures and heat transfer rates, as well as printed values of the integrated energy transfers Qaux and Qcoll (auxiliary energy rate and useful solar energy rate). The period of interest is January and February.

1.5.2. Modeling approach

Before starting creating the project in the Simulation Studio, we need to study the simulated system, decide what factors will be of interest and identify the components that will be used in the simulation.

The solar system consists of solar collectors, a pump, and an auxiliary heater. We also need to read and process weather data for Ouagadougou and output some results (both to the online plotter and to a printed file).

The following components will be used:

- Type 109 Weather data reader and processor: This component reads weather data and processes it to calculate the solar radiation properties on any surface.
- Type 3 Pump: Simple pump model (imposed flow rate)
- Type 1 Solar collector: adapted to flat-plate collectors with quadratic efficiency parameters.
- Type 24 Integrator: will be used to integrate energy rates
- Type 25: Printer
- Type 65: Online plotter

1.5.3. Step-by-Step instructions to create the Project

1.5.3.1. Creating an empty project

In the Simulation Studio, go to File/New and select Empty Project, then click on "Create". By default the project is created in %TRNSYS16%\MyProjects in a new folder that is named Project**n** where n is a number (it is omitted for the first project created, which is just called Project.tpf).

This will create an empty project. You can first set some global simulation parameters by going to "Assembly / Control Cards" or by clicking on the document icon as shown in Figure 1–23. You just need to change the simulation stop time so it displays 1416 (end of February) instead of the default value. If you scroll down in the Control Cards window, you will notice that the deck file is automatically given the same name as the project file with a .dck extension. You can change it if you wish.

1.5.3.2. Adding components and configuring them

You can now start adding components to the simulation.

WEATHER DATA READING AND PROCESSING

We will start with the data reader and solar radiation processor, Type 109. You can find it in the Direct Access Tool (tree structure in a docked window at the right of the screen). Type 109 is under "Weather data reading and processing". We need to read weather data for Ouagadougou (Burkina Faso, Africa), which is available as part of the weather database generated with Meteonorm (refer to Volume 9 of the documentation for more details on Weather Data). Meteonorm files use the TMY2 data format, so we will select the proforma for Type 109 that is in \Standard Format\TMY2. You can add the component to the project by dragging its icon from the direct access tool to the project window (see Figure 1–23).

Component configuration

- If you double-click on the component, you will have access to its parameters, inputs, outputs and other configuration settings like the external files. In our case we want to switch to the "External Files" tab and select the file for Ouagadougou, which is %TRNSYS16%\Weather\Meteonorm\Africa\BF-Ouagadougou-655030.tm2.
- Type 109 is also used to calculate the incident radiation on the solar collectors' plane so we need to set the slope and azimuth of the tilted surface appropriately. In the "Input" tab, set the Slope of the surface to 45 and the azimuth to 0 (0 means facing the equator, i.e. South in the Northern hemisphere). This will set the initial values of those inputs, and the initial value will be used throughout the simulation since we will not connect those inputs. In absence of information, you can leave the ground reflectance coefficient to 0.2.

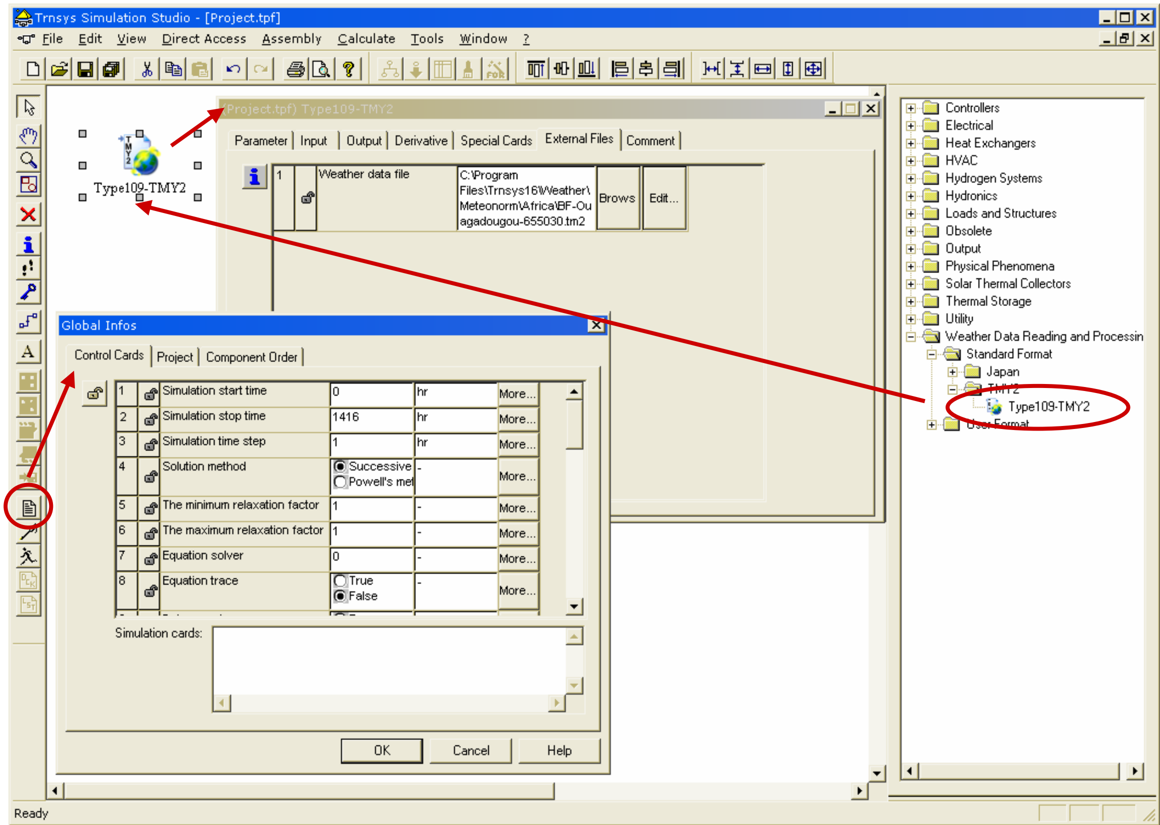

Figure 1–23: Control cards and Type 109 (Weather data reader / processor)

SOLAR SYSTEM

We will now add the 3 components required for the physical system model: Pump, Solar collector and Auxiliary heater. The selected components are:

- Hydronics\Pump\Single speed\Type3b
  - Changes to default configuration: set maximum flow rate (Parameter 1) to 50 kg/h
- HVAC\Auxiliary Heaters\Type6
  - Changes to default configuration: set maximum power (Parameter 1) to 5040 kJ/h = 1400 W
- Solar Thermal Collectors\Quadratic Efficiency Collector\2nd-Order Incidence Angle Modifiers\Type1b
  - Changes to default configuration: set the collector area (Parameter 2) to 2 m². Note that the given area is the total collector area, not the area of each module
  - No other changes are required if we accept some rounding errors on the efficiency parameters (the exact values in kJ/h are 12.996 kJ/h-m²-K and 0.0504 kJ/h-m²-K²). You do not need to change the initial value of the collector slope (Input 9) because it will be connected to the output of Type 109 (This will make it easier to change the collector slope later since only one value will need to be changed).

UTILITY COMPONENTS

We need to impose the flow rate on the pump by sending a time-varying control signal and we also need to integrate, plot and print the results. We will add 4 components:

- Utility\Forcing Functions\General\Type14h
  - Parameters: the profile we want to define is a step-like function that is 0 from 0 to 08:00, 1 from 08:00 to 18:00, and 0 again from 18:00 to 24:00 (the profile is repeated after 24h). I order to create that profile, we will need 6 points: (0;0), (8;0), (8;1), (18;1), (18;0), (24;0). So we should declare that we need 5 points besides the initial one, and enter the value of parameters as follows: 0; 0; 8; 0; 8; 1; 18; 1; 18; 0; 24; 0.

> The Simulation Studio offers a convenient way to define a forcing function using a graphical plug-in for Type 14. Please refer to section 1.5.3.3 for details.
>
> For step-like forcing functions, it is recommended to define the profile by repeating the time values with two different function values, as it is done here above, and to use the "average value of function" output of Type 14. This will ensure consistent results for any chosen time step

- Utility\Integrators\Quantity Integrator\Type24
  - The default configuration will integrate inputs from the beginning to the end of the simulation. We need to change the input tab to have 2 inputs (set "2" for "how many inputs should be integrated?"). This will automatically update the number of outputs.
- Output\Online Plotter\Online Plotter Without File\Type65d
  - Parameters: we will plot 3 variables on the left-axis, 2 on the right-axis. By setting the axis limits to [-100; 100] for the left (temperatures) and [0; 10000] for the right (heat transfer rates) we will ensure minimum overlapping of the plots.
- Output\Printer\No Units\Type25c
  - Without any changes to the default configuration, the printer will output results from the simulation start to the simulation end to a file named ***.out where *** is the deck file name (Project**n** if you did not rename it). You should change the number of printed outputs to 2 (answer to the question: " How many variables are to be printed by this component?" in the Input tab).

1.5.3.3. Connecting components

Components are connected using the Link tool, which is activated by pressing the link button (see Figure 1–24). When you move the mouse over a component icon, the 8 available connection points become visible. Click one of them to select the starting point, then go to the component you want to link and select a connection point again. Click to create the link. The newly created link is empty, i.e. it does not connect any (output; input) couple yet. This is shown by a different link color (blue by default, while links with connections are black).

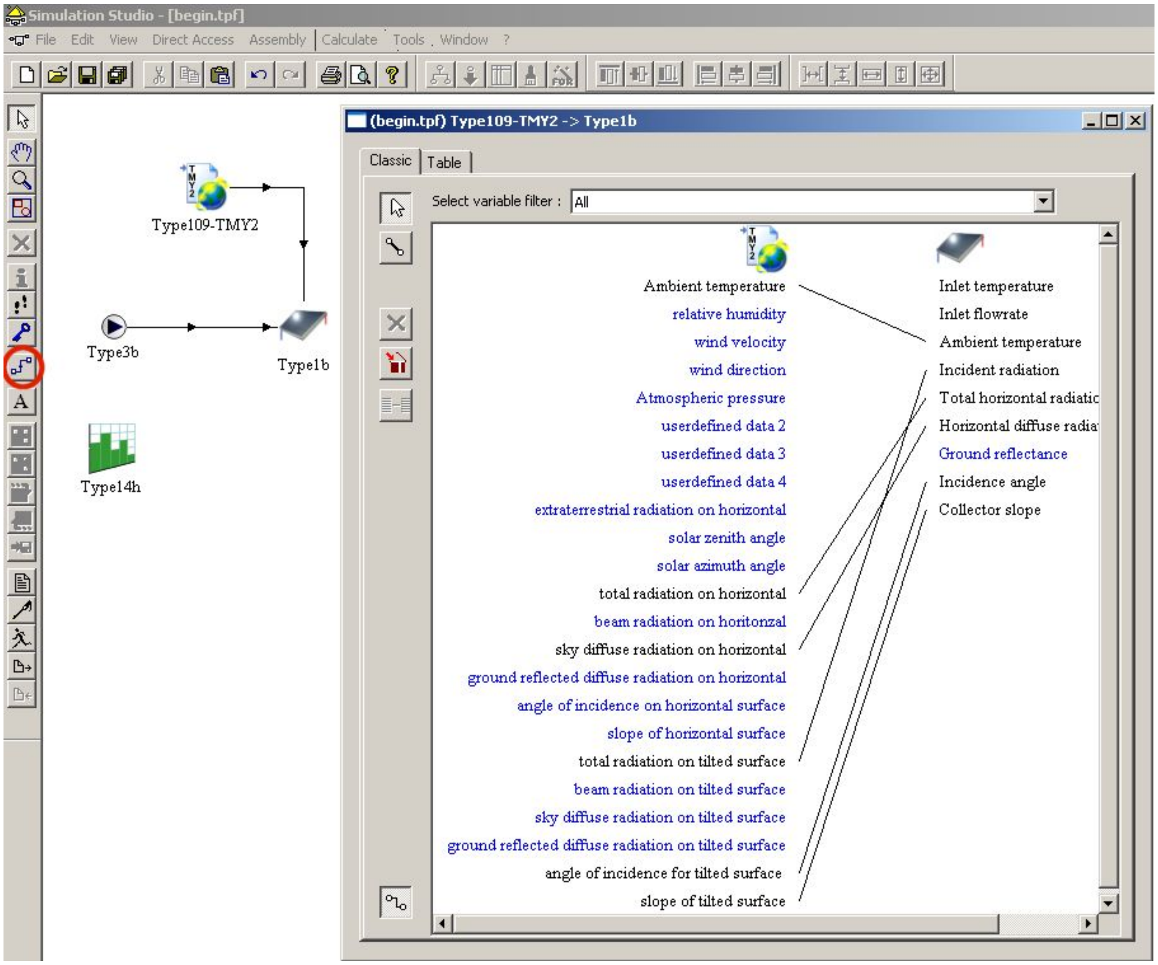

Figure 1–24: Connecting two components with the link tool

To connect outputs to inputs inside a link, you must go back to the selection tool by clicking on the white arrow icon or by pressing the ESC key. You can then double-click on the link and edit the connections as shown in Figure 1–24. You can use the filter and align tools and the alternative "table view" to make connections easier, as discussed in section 1.4.1.2.

The connections to be made in our case are listed here below:

SYSTEM CONNECTIONS

Type109 (Weather data) to Type1 (Solar collector)

- Ambient temperature → Ambient temperature
- Total radiation on horizontal → Total horizontal radiation
- Sky diffuse radiation on horizontal → Horizontal diffuse radiation
- Total radiation on tilted surface → Incident radiation
- Angle of incidence for tilted surface → Incidence angle
- Slope of tilted surface → Collector slope

Type 14 (Forcing Function) to Type 3 (Pump)

- Average value of function → Control signal

Type 3 (Pump) to Type 1 (Solar collector)

- Outlet fluid temperature → Inlet temperature
- Outlet flow rate → Inlet flow rate

Type 1 (Solar Collector) to Type 6 (Auxiliary heater)

- Outlet temperature → Inlet fluid temperature
- Outlet flow rate → Inlet flow rate

CONNECTIONS TO OUTPUT DEVICES

Connections to Type 24 (Integrator)

- Type 1 (Solar collector), Useful energy gain → Input to be integrated-1
- Type 6 (Auxiliary heater), Required heating rate → Input to be integrated-2

Connections to Type 65 (Online plotter)

- Type 3 (Pump), Outlet fluid temperature → Left axis variable-1
- Type 1 (Solar collector), Outlet temperature → Left axis variable-2
- Type 1 (Solar collector), Useful energy gain → Right axis variable-1
- Type 6 (Auxiliary heater), Outlet fluid temperature → Left axis variable-3
- Type 6 (Auxiliary heater), Required heating rate → Right axis variable-2

Connections to Type 25 (Printer)

- Type 24 (Integrator), Result of integration-1 → Input to be printed-1
- Type 24 (Integrator), Result of integration-2 → Input to be printed-2

Anytime during the process of adding components, if you press F2, the Studio will display the *Type number* and the *Unit number*. This option is very useful when you want to know this information about each component.

1.5.3.4. Setting labels

Some output components allow defining labels, or descriptors, for outputted variables. In the Simulation Studio, those variable descriptors are set in the input tab, instead of the initial value of inputs.

In our case, we need to define descriptors for Type 25 (Printer) and Type 65 (Online plotter). Figure 1–25 shows possible names for the printed and plotted variables.

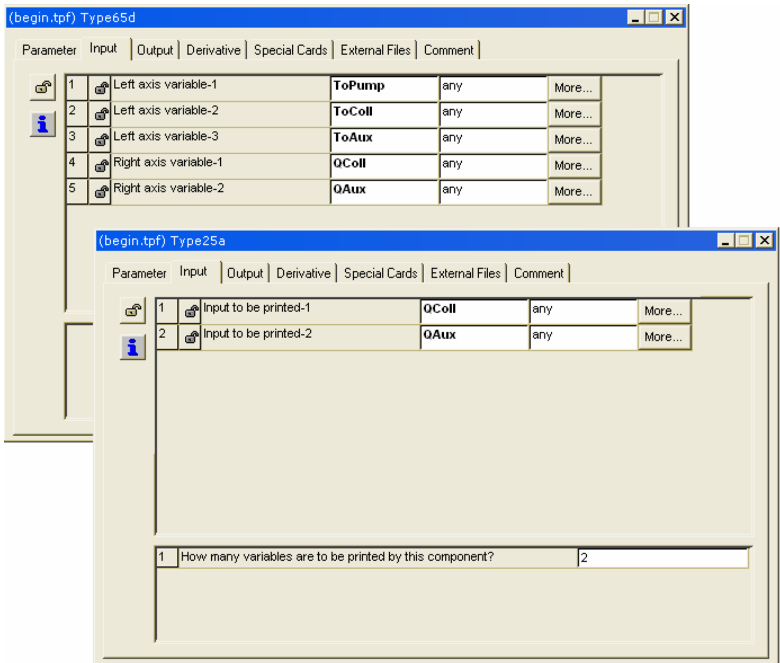

Figure 1–25: Defining variable descriptors for output components

1.5.3.5. Running the project and analyzing the results

Go to Calculate / Run Simulation or press F8 to run the simulation. The online plotter will display simulation results during the simulation and output files will be generated. If there is an error while processing the input file or running the simulation, you can view detailed error messages using the Error Manager (see "Troubleshooting a simulation" in section 1.4.1.3).

1.5.3.6. Some alternative options offered by the Studio

CONFIGURING TYPE14 (FORCING FUNCTION) USING ITS GRAPHICAL PLUG-IN

Type 14 is one of the standard components for which the plug-in technology has been implemented. In the Simulation Studio, any component can be associated with a separate executable program to help users setup its configuration. In the case of Type 14, the plug-in offers a graphical way of defining the time profile of the forcing function.

You can launch the plug-in by clicking on the wizard icon in the component's proforma. This will launch the external program, as shown in Figure 1–26. You can define the desired profile using the plug-in. When you are done, click OK and the plug-in will transfer the parameters to Type 14 and close.

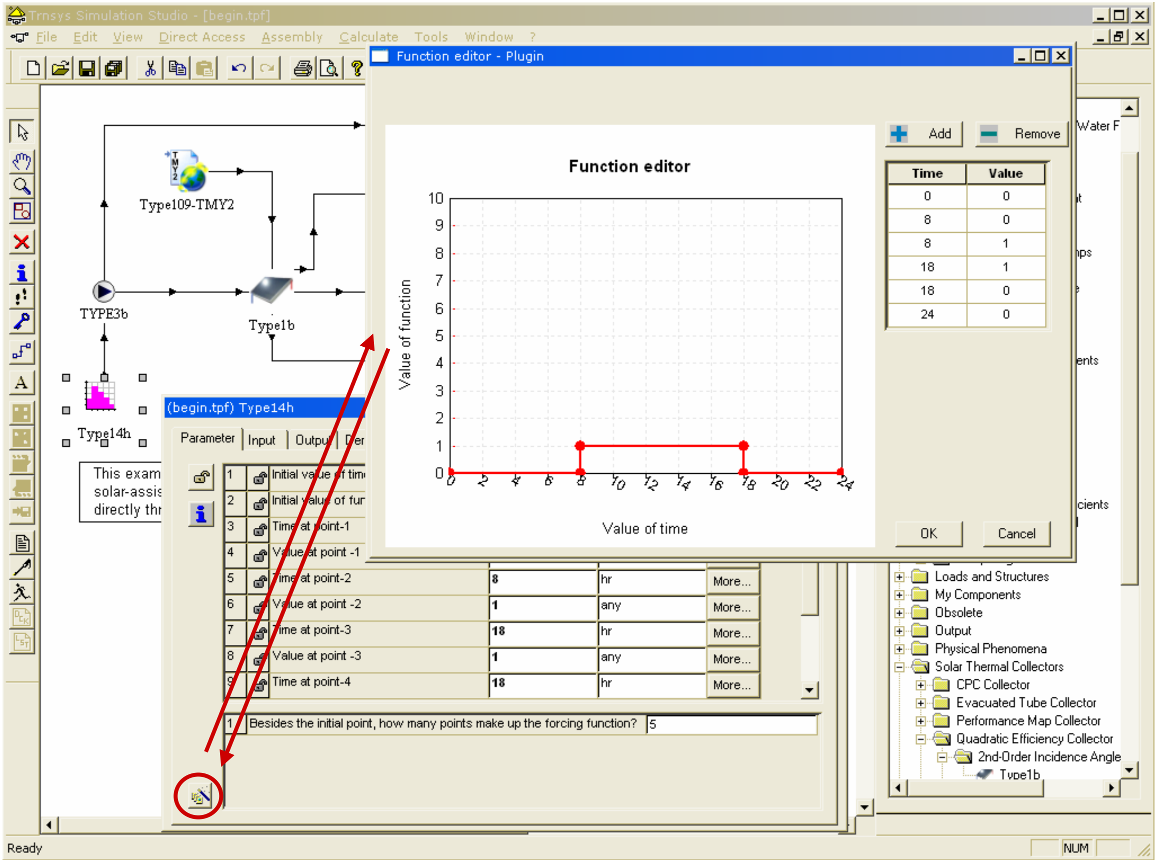

Figure 1–26: Control Using the Type 14 (Forcing Function) Plug-in

DEFINING PLOTTED AND PRINTED VARIABLES USING THE OUTPUT MANAGER

In the step-by-step instructions here above, we have added an online plotter and a printer to the simulation, as we added other components. The Simulation Studio offers another way to plot and print variables: the Output Manager.

The Output Manager is launched using the "Assembly" Menu. Figure 1–27 shows how an online plotter with the same connections as the ones described here above can be defined:

- Add an online plotter (online plotter icon on the right)
- Click on the created online plotter in the tree structure on the right and configure it to have 3 left-axis variables and 2 right-axis variables. You can also change axis limits, etc. here.
- Expand the online plotter tree on the right and the desired component (e.g. pump) tree on the left, select the output to be connected to the first plotted variable (outlet fluid temperature), select the "left-axis variable–1" of the online plotter, and click on the right arrow to make the connection. Repeat for all other variables to be plotted.

When you are done, close the output manager. A new online plotter named "System_Plotter" has been created in the project. You can modify the plotter parameters and connections indifferently in the project window or in the output manager. In particular, you can change the variable label ("name") in the "input" tab of the proforma.

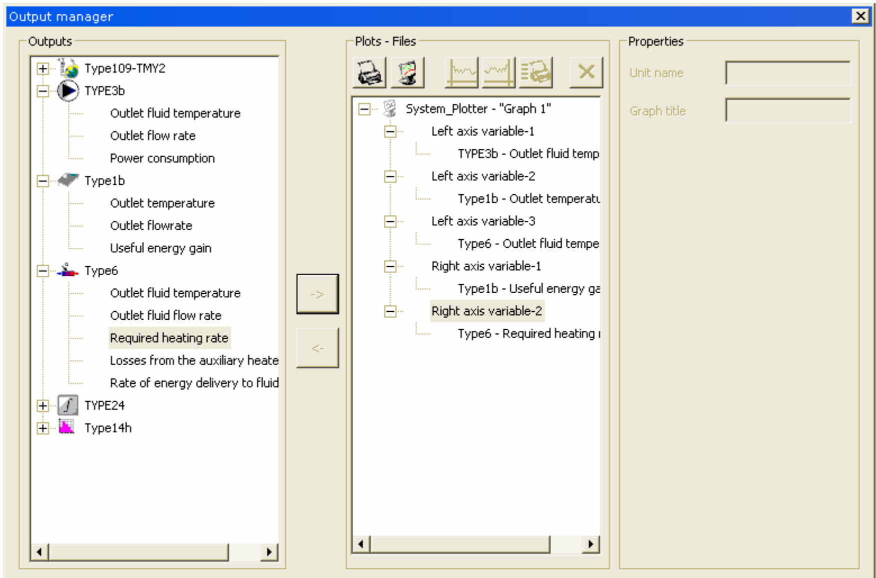

Figure 1–27: The output manager

For output files, there is even a shorter way to send outputs to a new printer: for any component, double-click on its icon and go to the "output" tab. Next to each output is a checkbox. If you check one variable, it will be automatically sent to a system printer, which is created the first time an output is selected. This is illustrated in Figure 1–28 for a solar collector (the first output would be sent to the default printer). You can then go to the output manager or to the System Printer icon to change some parameters like the start and stop time for printing, the filename, etc.

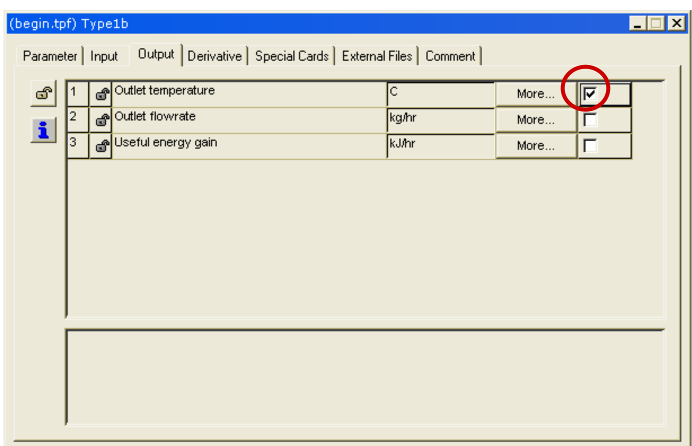

Figure 1–28: Output tab of a proforma and checkboxes to print any output

1.6. Creating a building project

This section presents step-by-step instructions to create a simple building project using the TRNSYS Studio Building Wizard. For detailed instructions on how to input building data using TRNBuild, please refer to Volume 6 of the documentation.

1.6.1. Using the Building wizard

The building wizard is described in detail in Volume 2 (Using the Simulation Studio). Please refer to that manual for further information. The next paragraph will only give step-by-step instructions to generate a simple building project.

To launch the building wizard, choose File / New and then select "Building Project (Multizone)"

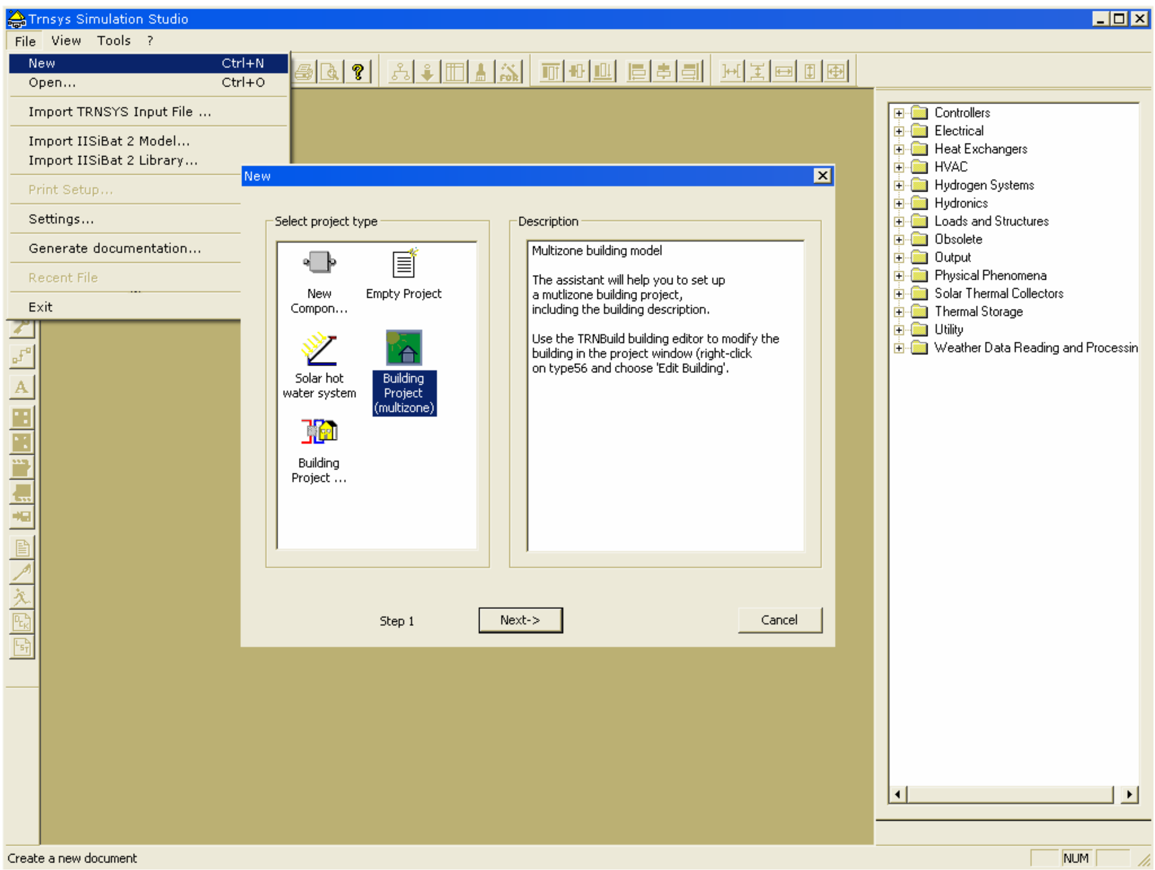

Figure 1–29: Building Wizard Step 1: Launching the building wizard

The first step in defining the building is to create thermal zones and tell the wizard which zones are adjacent to which others. This is done using a grid layout. Clicking in a cell of the grid will create a zone at that location. In this example, we create 3 zones in E5, E6 and F6.

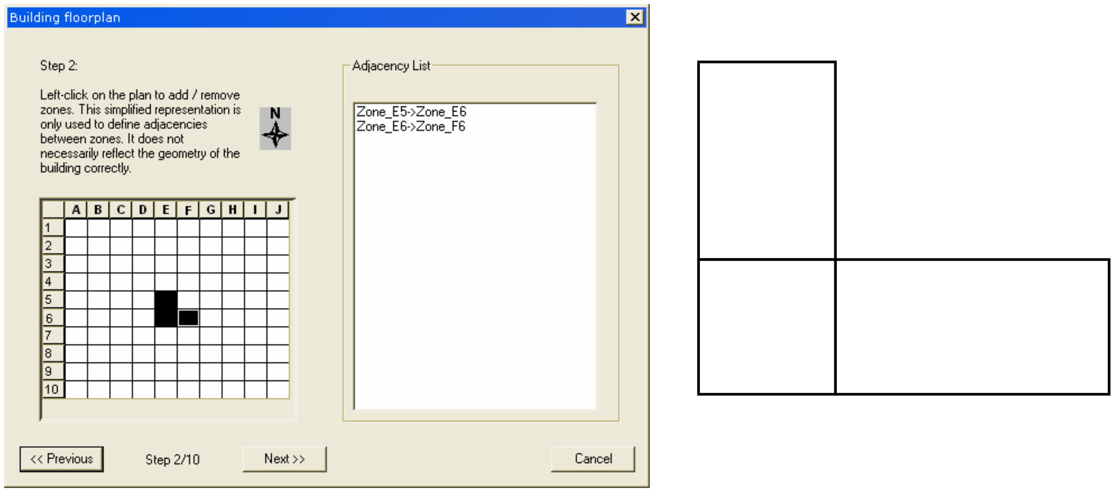

Figure 1–30: Building Wizard Step 2: Zones layout

The representation uses squares independently of the real shape of the building. In our case, the real shape of the building (which can be refined later in TRNBuild) is different from the simplified "square grid" view.

The 3rd step is to enter the dimensions of each room:

- Zone E5: Width = 6, Depth = 9, Height = 3
- Zone E6: Width = 6, Depth = 6, Height = 3
- Zone F6: Width = 12, Depth = 6, Height = 3

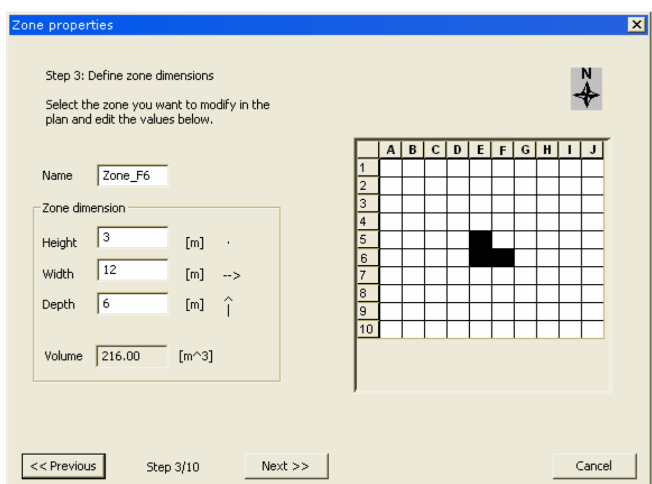

Figure 1–31: Building Wizard Step 3: Room dimensions

We can then set the glazing fraction of each side of the building (to be distributed equally among all zones) and a global rotation angle for the building (with respect to due North). The weather data file also selected. All those parameters can be changed later, once the project is created. The building in this example has 10% glazing fraction for the North side, 25% for East and West and 50% for South. "North" is actually 25° East of North.

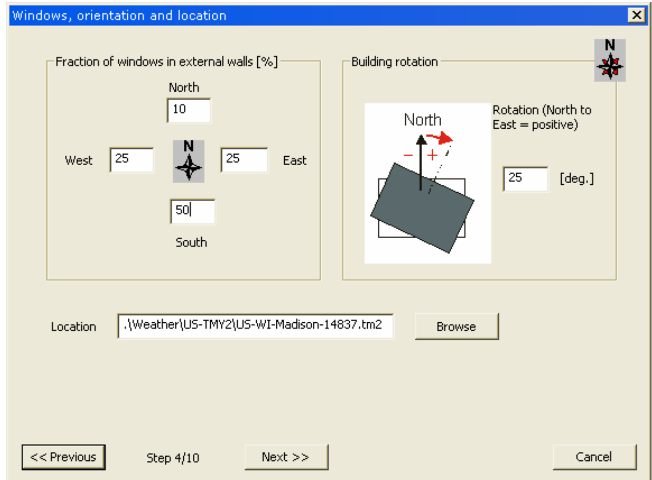

Figure 1–32: Building Wizard Step 4: Building orientation and glazing

We can now define the infiltration and ventilation. In this example, infiltration is 0.2 vol/h at all times and ventilation is 1 vol/h when the building is occupied (0 else).

Note: the building uses a default schedule for occupancy. All schedules can be modified later in TRNBuild)

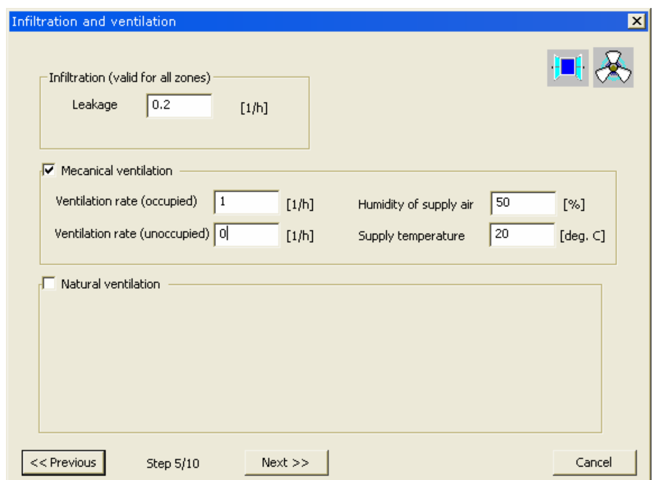

Figure 1–33: Building Wizard Step 5: Infiltration and ventilation

This building has heating and cooling. Set points, maximum power and other characteristics are defined in step 6. We will use the default values except for the cooling set point (constant value of 26°C).

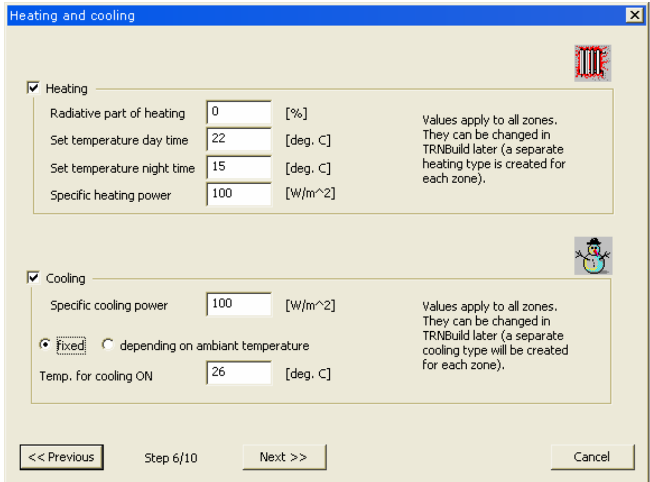

Figure 1–34: Building Wizard Step 6: Heating and Cooling

Internal gains use the default values.

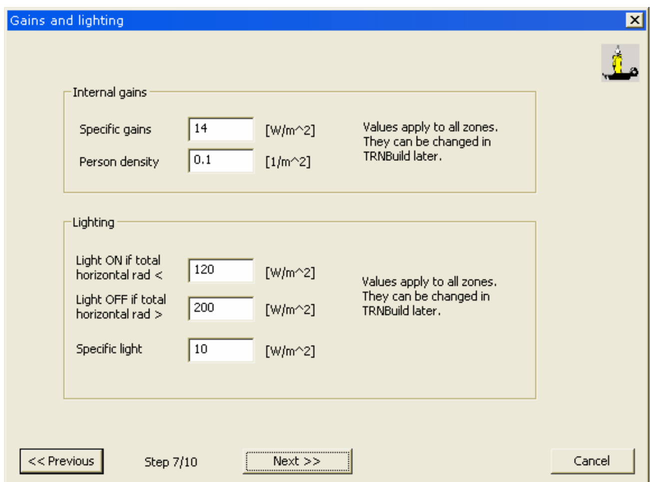

Figure 1–35: Building Wizard Step 7: Internal gains

The last two steps of the building wizard allow us to define shading devices. Step 8 is used to define fixed shading devices (overhangs and wingwalls). We only define an overhang on the South side of the building. The dimensions of the window (height = 2 m, width = 13.5 m) approximate the real windows as one big window. We assume that equivalent window is shaded by a 0.5-m overhang that is attached 0.1 m above the window.

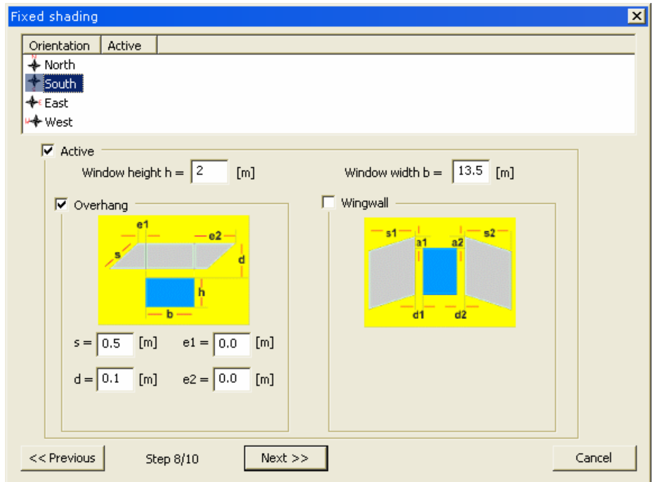

Figure 1–36: Building Wizard Step 8: Fixed shading

Movable shadow is only used on the South side of the building as well. We keep the default values for thresholds and shaded coefficients.

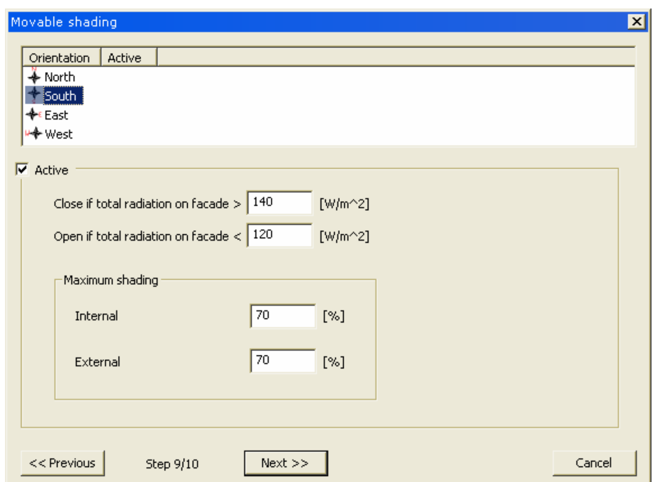

Figure 1–37: Building Wizard Step 9: Movable shading

We can now generate the TRNSYS project. The building wizard will generate a TRNSYS Project and a Building Description file (.bui). It calls TRNBuild to generate the associated building files, and then opens the created project in the Studio.

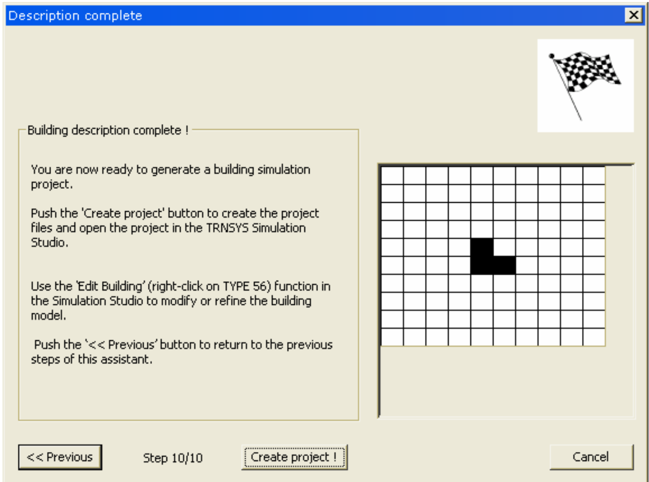

Figure 1–38: Building Wizard Step 10: Generating the TRNSYS Project

The TRNSYS project created by the building wizard has all the necessary components and the connections have been created for you. You can explore them by double-clicking on any component or link.

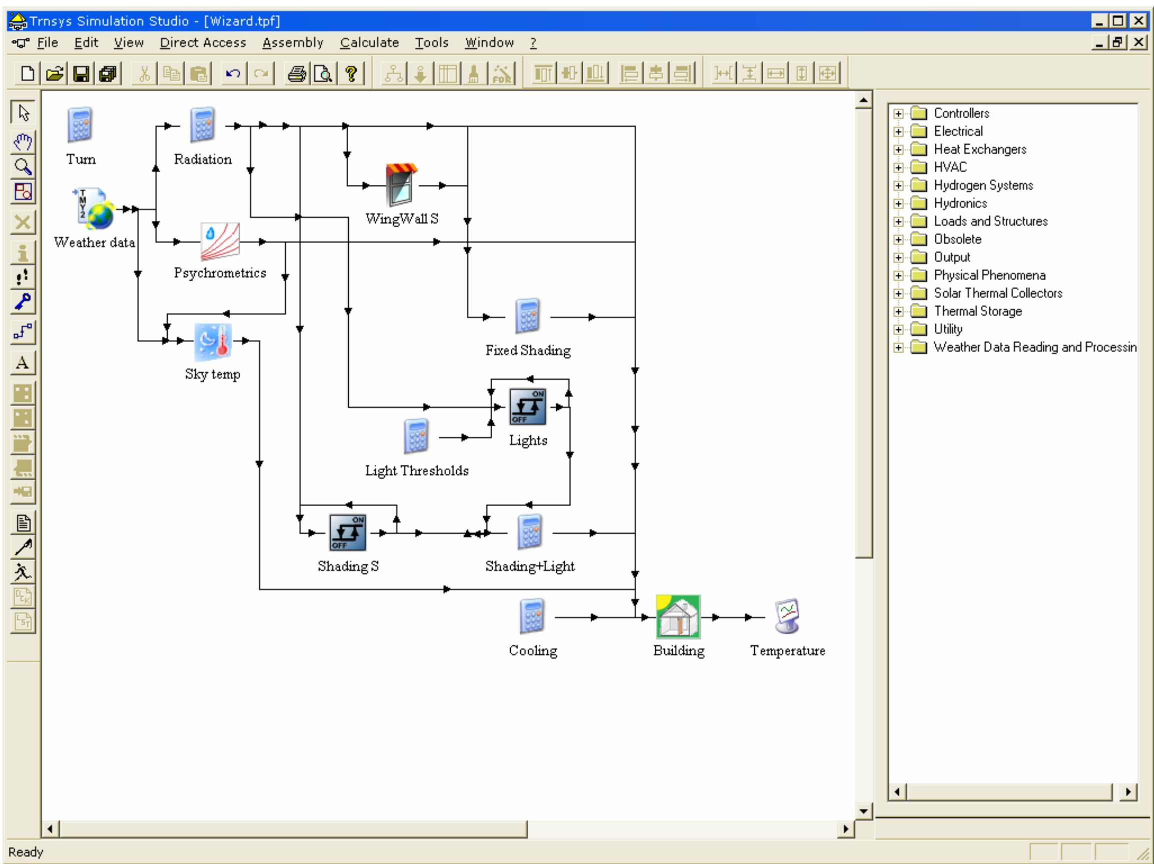

Figure 1–39: TRNSYS Project generated by the Building wizard

1.6.2. Modifying the wizard-generated project

This section will briefly illustrate how you can modify the project generated by the building wizard. You can modify the building itself but also change or add components in the simulation.

First, we will shorten the simulation period: you can run the simulation for the first week of July (hours 4344 to 4512) by changing the "Start" and "Stop" times in global control cards (accessed through Assembly/Control Cards or using the document icon circled in Figure 1–40.

You can change the online plotter axis limits by double-clicking on its icon and changing the parameters (see in Figure 1–40). Set the temperature axis limits of the online plotter to 20 and 40.

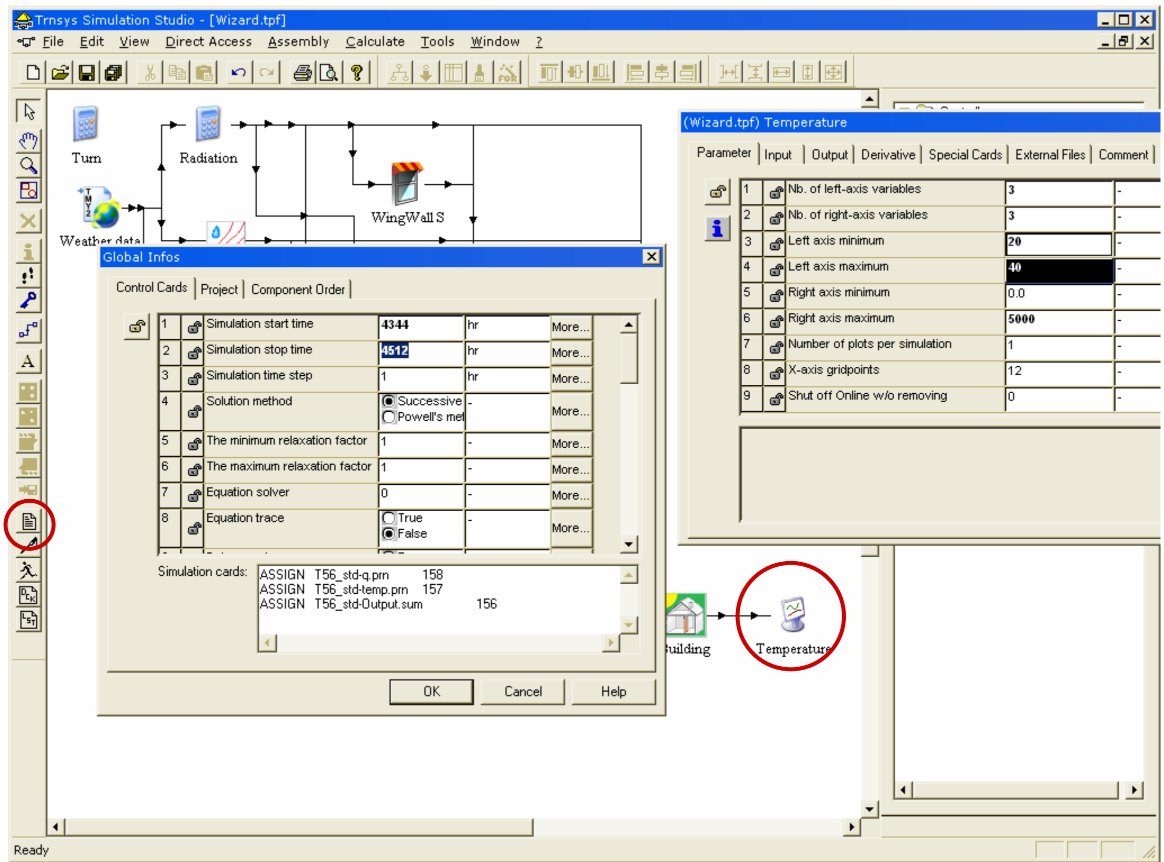

Figure 1–40: Changing the simulation start and stop and the online plotter configuration

Turn cooling off by opening the Equation Editor for "Cooling" and setting the "T_COOL_ON" variable to 50 °C (see Figure 1–41).

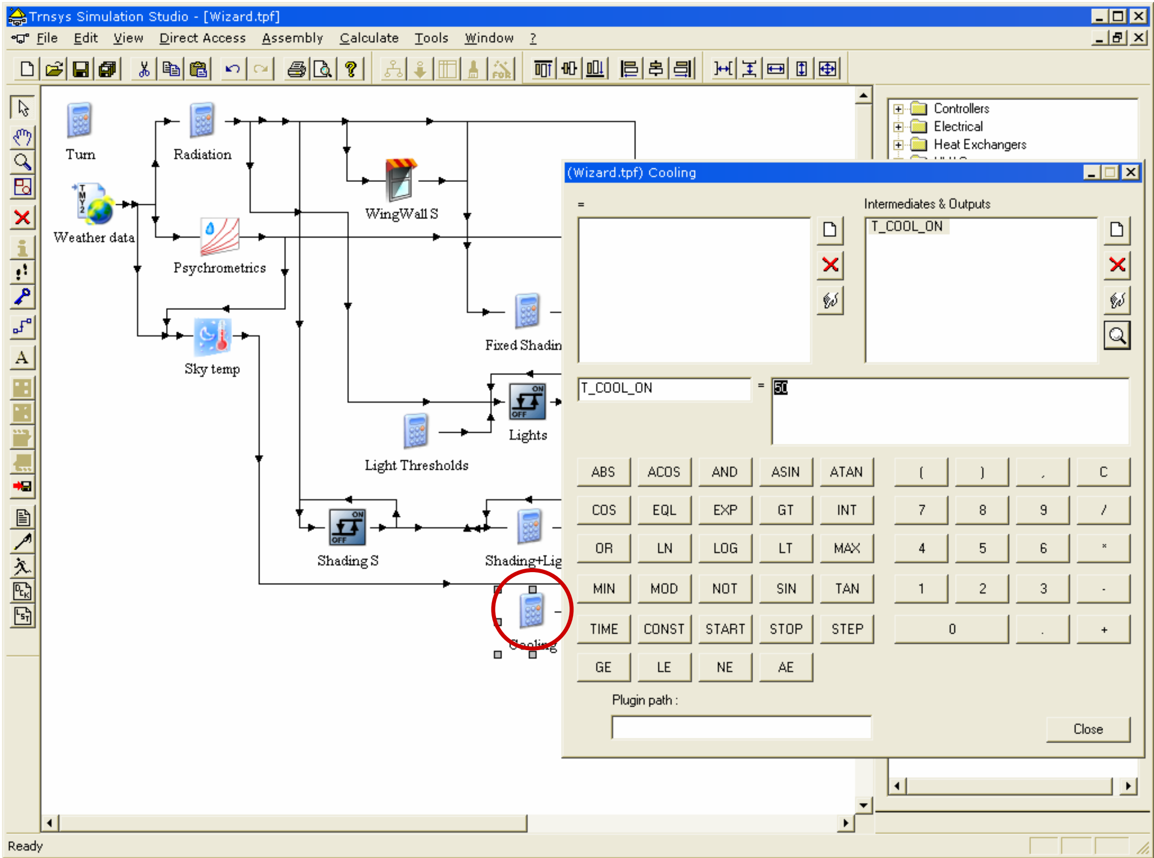

Figure 1–41: Turning off cooling

Run the simulation and take a look at the plotted temperatures (see Figure 1–42).

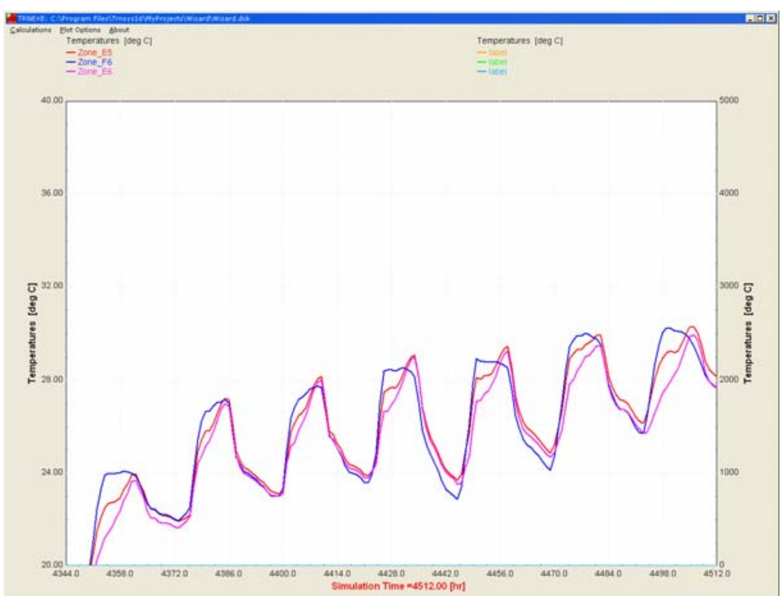

Figure 1–42: Temperatures for the first week of July

Now we are going to make some changes to the building and study the influence on the results. To edit the building, right-click on the Building icon and selecting "Edit Building". This will launch TRNBuild (see Figure 1–43).

- Select Zone F6.
- The South wall is of Type OUTWALL and has an area of 36 m² (including 18 m² of glazing). Its orientation is SHADSOUTH.
- Change the wall orientation to SOUTH. This will change the incident radiation for the wall and the included windows, effectively removing the overhang and movable shading effects.
- Change the wall area to 35 m².

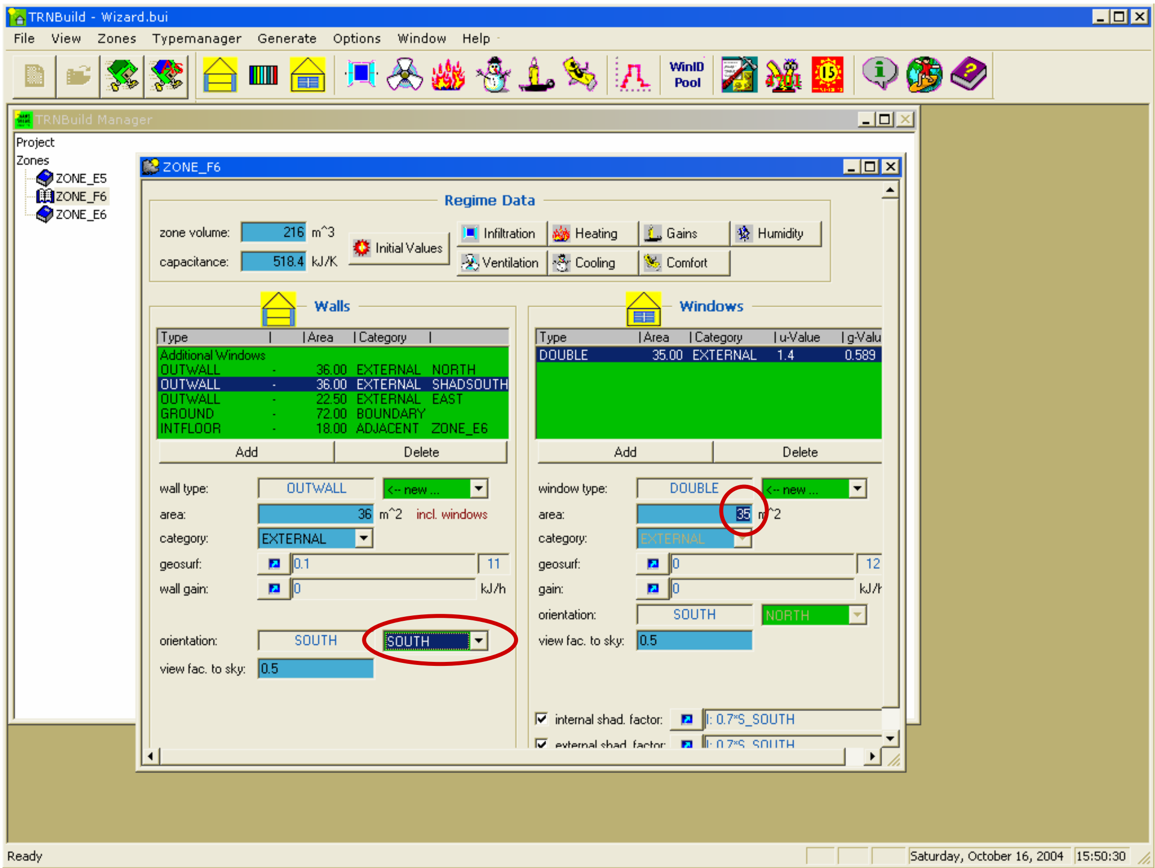

Figure 1–43: Wizard-generated building in TRNBuild

Save the file. Go back to the TRNSYS Studio by closing TRNBuild and run the simulation again (press F8). The temperature in zone F6 increases dramatically, as much more solar gains enter the zone (larger window area and no shading). Figure 1–44 shows both results on the same graph (This is an edited picture; it is not possible to show successive runs simultaneously in the online plotter).

Figure 1–44: Temperatures before and after changing the South wall / window in zone F6

1.7. Using TRNEdit and creating a distributable (TRNSED) application

This section explains how to create a simple TRNSYS-based redistributable application using the Simulation Studio and TRNEdit / TRNSED.

> Note: Redistributable stand-alone applications based on TRNSYS, known as TRNSED Applications, are subject to a special license agreement. The basic terms of the agreement is that you have the right to distribute those applications free of charge and that you have to negotiate a contract with the TRNSYS developers if you want to sell those applications. Please contact your TRNSYS Distributor if you have questions about licensing.
>
> Some distributors may not activate the "Create TRNSED" function by default. If it is the case, the "TRNSED/Create distributable" menu item will be disabled in TRNEDit. You should contact your distributor to activate it.

1.7.1. Starting point: TRNSYS Studio project

Open Examples\TRNSED\SDHW-TRNSED (Studio).tpf. This is a slightly modified version of the SDHW example. The modifications are listed here below. They are already made in the project distributed with TRNSYS. If you want to repeat the whole process, please replace Examples\TRNSED\SDHW-TRNSED (Studio).tpf with a fresh copy of Examples\SDHW\SDHW.tpf

Set "Write TRNSED Commands" to "true" and change the deck file name in Control Cards. The deck filename should say: SDHW-TRNSED (Studio).**trd**. TRNSED applications must have a **.trd** extension.

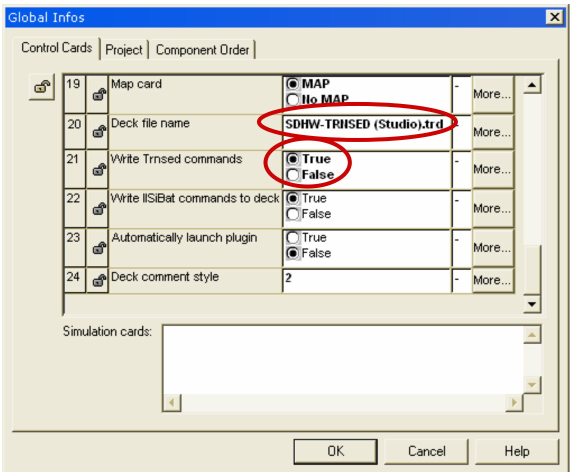

Figure 1–45: Modified Control Cards for TRNSED projects

TRNSED will present a simplified view of the projects, with only some of the parameters available to users. You need to select those parameters by keeping them "unlocked", while you "lock" all other parameters to hide them.

This is done by opening the proformas of all components in the simulation and clicking on the lock icon for the corresponding parameters. It is also possible to lock all parameters of one component by clicking on the lock icon in the upper left corner of each tab. You must lock **all** fields in the Studio that you don't want users to be able to change. This includes parameters but also initial values for inputs and derivatives, and filenames (see Figure 1–46 for an example).

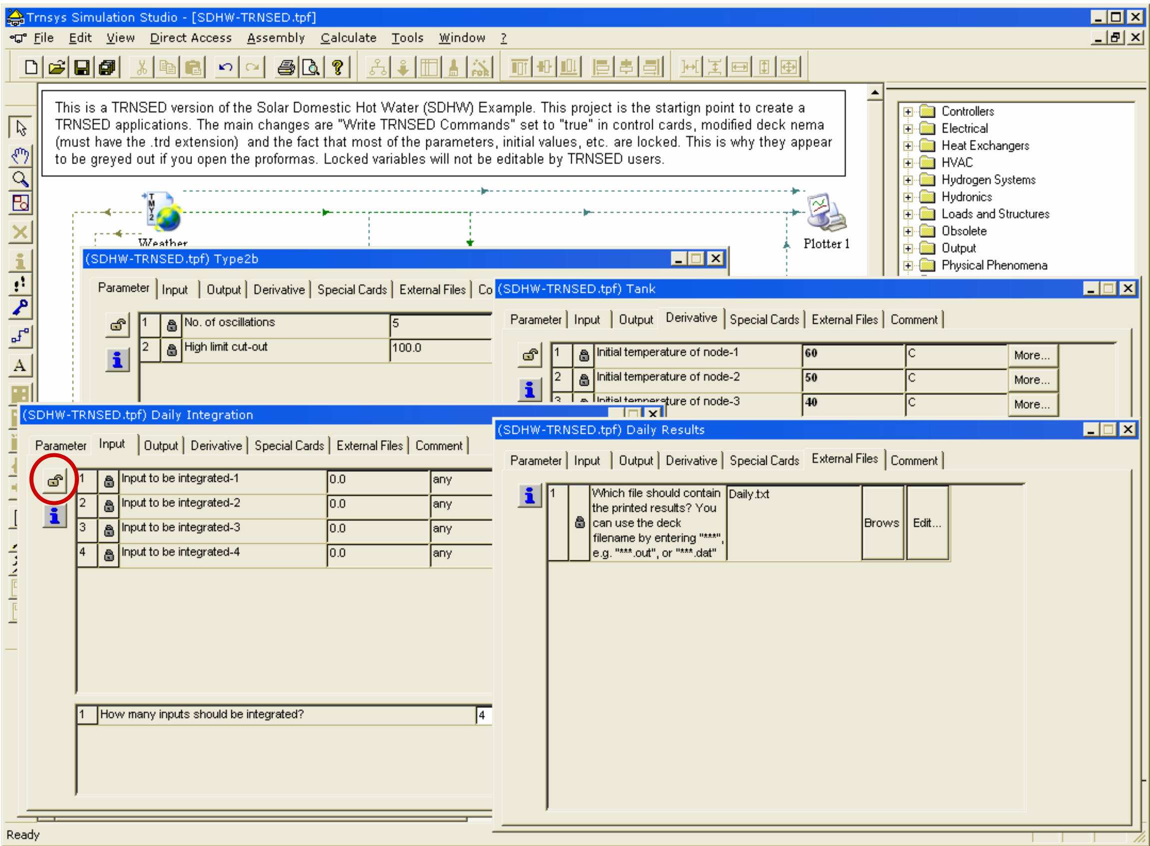

Figure 1–46: Locking parameters, initial input values, file names, derivatives initial values

In this case, we will let users change the weather data file used, the solar collector slope and azimuth (Set in Type 109, the weather data processor), the solar collector area, the tank volume and the set points for auxiliary heating and for domestic hot water after the mixing valve. All other input fields should be locked (not all windows are shown in Figure 1–46, you need to double-click on every single component in the simulation).

Once all the "lock-unlock" settings are correct, you can generate the TRNSYS input file (the TRNSED deck in this case) by using "Calculate / Create input file" or using the "pen" button.

Note: This will re-create SDHW-TRNSED (Studio).trd, which you can use as a working copy for the following. The final TRNSED file after editing is saved in SDHW-TRNSED.trd. You can use this second file to examine the changes without actually performing them

1.7.2. Editing the TRNSED file in TRNEdit

After creating the file, launch TRNEdit and open it. TRNEdit recognizes that the input file is a TRNSED file by its extension and by the "*TRNSED" command that is included by the Studio. TRNEdit creates two tabs, one displaying the source code of the input file, the other displaying the TRNSED view of the file. Both tabs are shown in Figure 1–47.

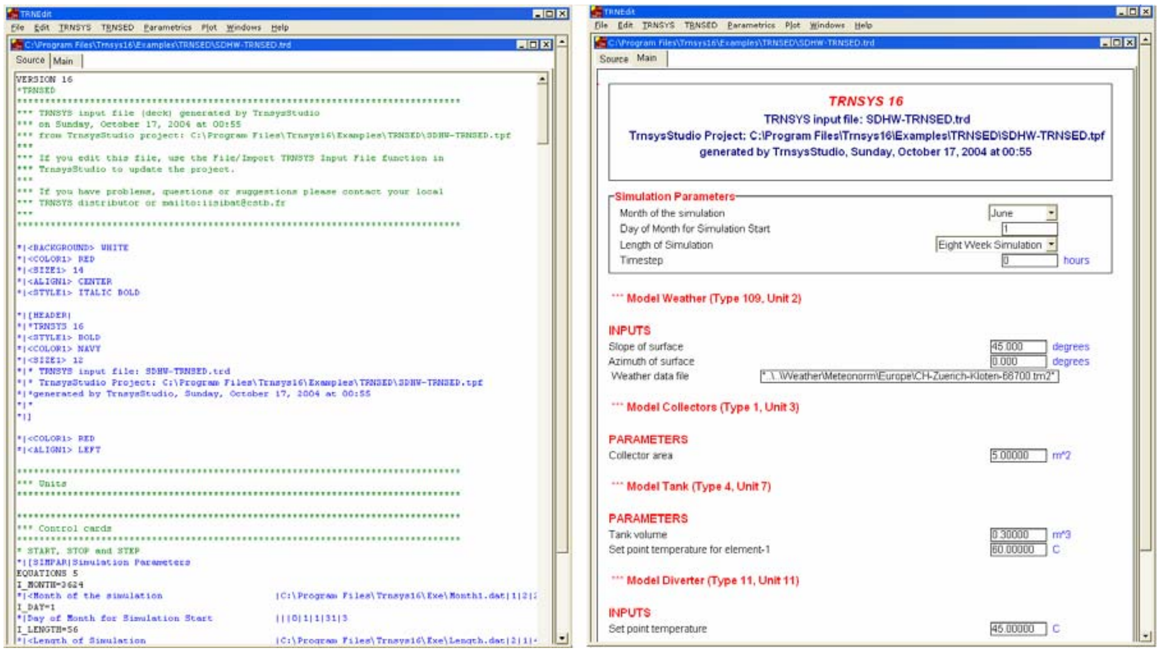

Figure 1–47: Opening the input file in TRNEdit: the Source tab and the TRNSED view

By flipping back and forth between the two tabs, you can immediately see the impact of your changes to the source code in the TRNSED view.

1.7.2.1. Rules for editing TRNSED files

All TRNSED statements start with " *| ". The first character (*) makes sure that TRNSYS will ignore those instructions (lines starting with a * and all characters after an exclamation mark are ignored by TRNSYS).

The first few lines of the TRNSED file set the title and first comment block. You can edit it to match the following lines (modified text is in bold):

```
*|<BACKGROUND> WHITE
*|<COLOR1> RED
*|<SIZE1> 20
*|<ALIGN1> CENTER
*|<STYLE1> ITALIC BOLD

*|[HEADER|
*|*Solar Domestic Hot Water System
*|<STYLE1> BOLD
*|<COLOR1> NAVY
*|<SIZE1> 12
```

```
*|* TRNSYS input file: SDHW-TRNSED.trd
*|* A simple TRNSED demo
*|]
```

A few TRNSED instructions illustrated here are:

- Groups: They are surrounded by a black border to identify a group of settings that go together but they also serve other purposes (e.g. a group can be turned on or off by TRNSED controls). A group has a name for TRNSED controls (no blanks) and a title that is displayed in the TRNSED view (if that name is blank, no title is displayed, as it is the case here). A group starts with:
  `*|[GroupName|Group title`
  and ends with
  `*|]`
- Text Style properties such as *|<COLOR1> RED. This command sets the color of comments. You can use usual color names or any color by specifying values for Red, Green and Blue levels (1 to 255) with the following syntax (the numbers here below will result in the darker red used in the SEL logo):
  `*|<color1> rgb(204,0,0)`
  Other instructions are *|<STYLE1> (e.g. bold, italic), *|<ALIGN1> (left, center, right).
- Comments: They start with *|*. The text after that is just displayed by TRNSED

We can create additional groups in the file (e.g. one for each component). The section about Type 109 (Data reader and weather data processor) is

```
*|*
*|* *** Model Weather (Type 109, Unit 2)
*|*
CONSTANTS 2
*|*INPUTS
SLOPEOFS=45
*|Slope of surface                          |degrees|degrees|0|1|0|90.000|1000
AZIMUTHO=0
*|Azimuth of surface                        |degrees|degrees|0|1|-
360|360.000|1000

UNIT 2 TYPE 109     Weather
*$UNIT_NAME Weather
*$MODEL .\Weather Data Reading and Processing\Standard Format\TMY2\Type109-
*$POSITION 103 129
*$LAYER Weather / Data Files # Weather - Data Files #
PARAMETERS 4
2             ! 1 Data Reader Mode
36            ! 2 Logical unit
4             ! 3 Sky model for diffuse radiation
1             ! 4 Tracking mode
INPUTS 3
0,0           ! [unconnected] Ground reflectance
0,0           ! [unconnected] Slope of surface
0,0           ! [unconnected] Azimuth of surface
*** INITIAL INPUT VALUES
0.2 SLOPEOFS AZIMUTHO
*** External files
ASSIGN "..\..\Weather\Meteonorm\Europe\CH-Zuerich-Kloten-66700.tm2" 36
*|? Weather data file |1000
```

You can edit it to remove unnecessary comments inserted by the Studio (Those comments are useful if the deck file was to be re-imported in the Studio) and to enclose all input fields related to this component in a group called "LOCATION".

```
* Weather data file and Collector azimuth and slope
*|[LOCATION|Location
CONSTANTS 2
SLOPEOFS=45
*|Slope of surface                        |degrees|degrees|0|1|0|90.000|1000
AZIMUTHO=0
*|Azimuth of surface                      |degrees|degrees|0|1|-
360|360.000|1000

UNIT 2 TYPE 109      Weather
PARAMETERS 4
2             ! 1 Data Reader Mode
36            ! 2 Logical unit
4             ! 3 Sky model for diffuse radiation
1             ! 4 Tracking mode
INPUTS 3
0,0           ! [unconnected] Ground reflectance
0,0           ! [unconnected] Slope of surface
0,0           ! [unconnected] Azimuth of surface
*** INITIAL INPUT VALUES
0.2 SLOPEOFS AZIMUTHO
*** External files
ASSIGN "..\..\Weather\Meteonorm\Europe\CH-Zuerich-Kloten-66700.tm2" 36
*|? Weather data file |1000
*|]
```

After making similar changes to include all individual components in groups, you will get the TRNSED view shown in (The result after all the editing work is saved in Examples\TRNSED\SDHW-TRNSED.trd)

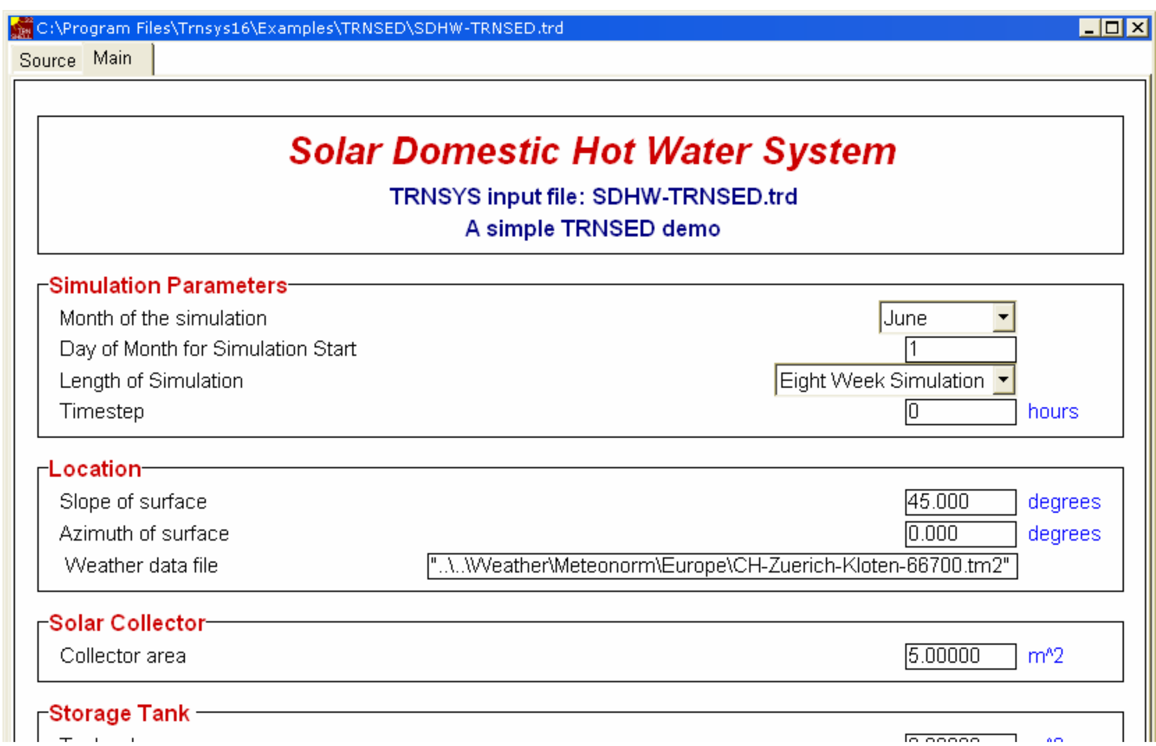

Figure 1–48: TRNSED view after creating groups

Note: If you forgot to "lock" some parameters in the Simulation Studio or if you want to hide additional parameters, it is easy to remove the TRNSED Statements without modifying the TRNSYS statements by deleting the lines starting with "*|" or commenting them out. This is easily done by adding a second "*" in front of the "|". The statements here below will display an input field for the Collector slope and another one to select the weather data file:

```
SLOPEOFS= 4.5000000000000E+01
*|Slope of surface                          |degrees|degrees|0|1|0|90.000|1000
…
ASSIGN "..\..\Weather\Meteonorm\Europe\CH-Zuerich-Kloten-66700.tm2" 36
*|? Weather data file |1000
```

The modified version here below is the same as far as TRNSYS is concerned but will not create the TRNSED input fields:

```
SLOPEOFS= 4.5000000000000E+01
**|Slope of surface                          |degrees|degrees|0|1|0|90.000|1000
…
ASSIGN "..\..\Weather\Meteonorm\Europe\CH-Zuerich-Kloten-66700.tm2" 36
**|? Weather data file |1000
```

1.7.3. Some refinements

1.7.3.1. Adding an input field

There are a few things we can do to improve the usability of our TRNSED demo:

We can add a parameter that is the daily load. That parameter is set by an equation block in the simulation and the Studio did not create an input field for it.

Search for the following lines in the file (Edit/Search "Daily load" for example):

```
* EQUATIONS "Daily load"
EQUATIONS 2
mdDHW = [9,1] * 200        ! Multiply by daily consumption
TCold = 15
```

We will replace the constant 200 with a new variable, DayDraw:

```
mdDHW = [9,1] * DailyDraw        ! Multiply by daily consumption
```

Then we have to define that variable and include it in a TRNSED command in the "DHW Load" group:

```
*|[DHW|DHW Load
Constants 1
DailyDraw = 200
*|Daily hot water draw   |l/d|m^3/d|0|0.001|0|1000.00|9999
… Rest of the DHW group
*|]
```

Do not forget the last line that closes the group.

The syntax to add an input field is as follows:

TRNSYS CONSTANT or EQUATION (`Variable = 12345.6789`)

```
*|Description   |Units1|Units2|Add|Mult|Min|Max|Help
```

With:

- Description: Descriptive text displayed by TRNSED
- Unit1: Variable units in the primary unit system
- Unit2: Variable units in the secondary unit system
- Add and Mult: Unit conversion between 1 and 2: Unit 2 = Add + Mult*Unit1
- Min: Minimum acceptable value for the parameter
- Max: Maximum value of the parameter. This number also fixes the format used to display the variable. If you enter 100 for the maximum and want to display 3 digits after the decimal point, you need to specify 100.000
- Help: Help number in a text file that must have the same name as the .trd file with an .hlp extension (not used here)

After adding that line, the DHW load group now looks like Figure 1–49.

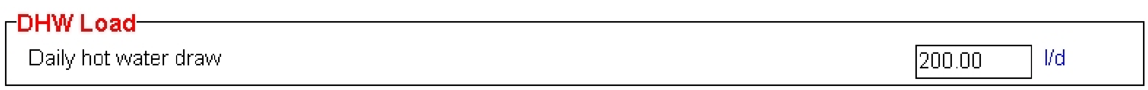

Figure 1–49: Adding an input field

1.7.3.2. Reorganizing TRNSED fields

We could consider removing the "Location" Group. The weather data file choice would move to the "Simulation Parameters" group and the collector slope and azimuth would move to the "Solar collector" group.

Note: When moving TRNSED commands around, it is important to remember that the file has to be run by TRNSYS. It is important to comply with the TRNSYS syntax when modifying a TRNSED input file

The initial view is shown in Figure 1–50

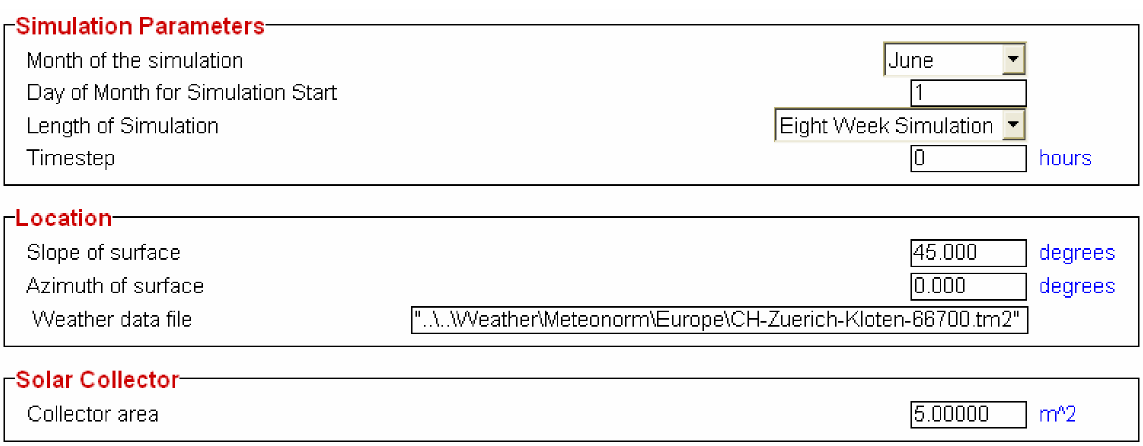

Figure 1–50: Reorganizing the input file (1)

To modify the file:

Move the opening command for the "solar collector" group up and replace the existing opening of the "Location" group with it. Delete the closing statement (*|]) for group "Location". The result is shown in Figure 1–51.

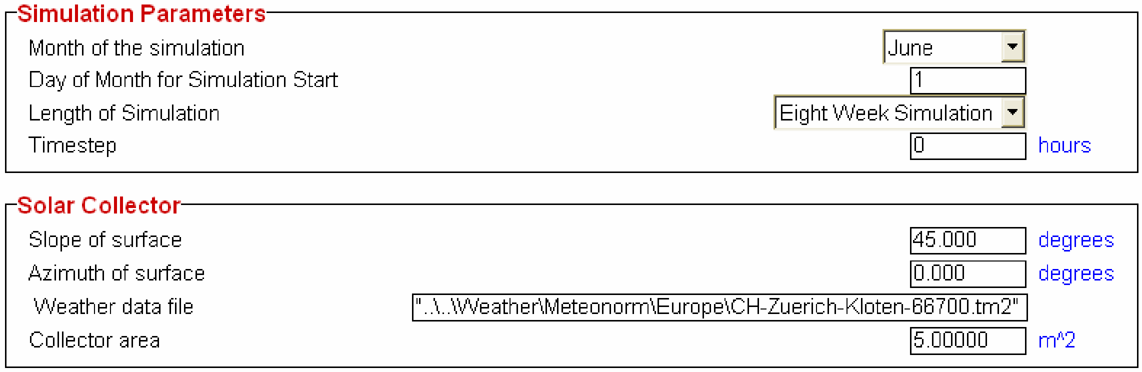

Figure 1–51: Reorganizing the input file (2)

Move the commands that allow users to select the input file up, into the "Simulation Parameters" group. The two lines making the statement are:

```
ASSIGN "..\..\Weather\Meteonorm\Europe\CH-Zuerich-Kloten-66700.tm2" 36
*|? Weather data file |1000
```

You must move both lines together. Note that TRNSYS will associate the file correctly with the data reader thanks to the logical number unit (36) which is among the parameters of Type 109. Move both lines just above the closing statement for the "SIMPAR" group. It may also be a good idea to prevent users from changing the simulation time step by adding a "*" in front of the line that defines the input field: "*|Timestep …" becomes "**|Timestep …". The result is shown in .

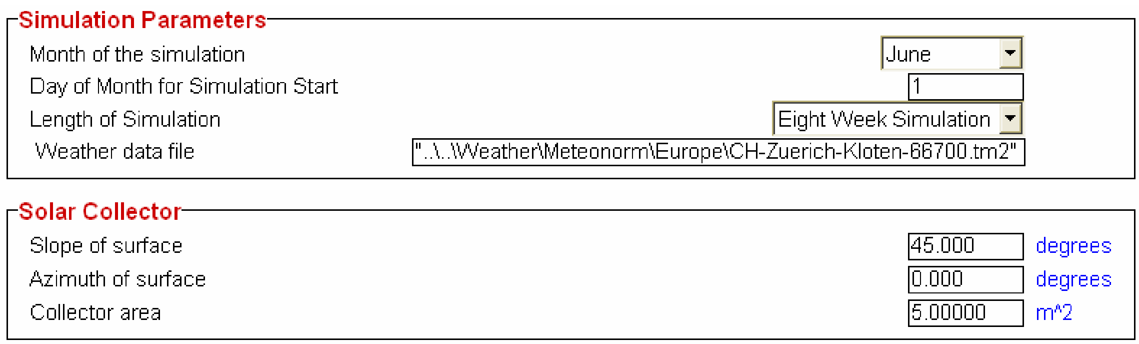

Figure 1–52: Reorganizing the input file (2)

1.7.4. Adding pictures, links and multiple tabs

1.7.4.1. Pictures

We will first add system schematics to the file to make it more user-friendly. TRNSED can use bitmaps (.bmp files). The Examples\TRNSED folder contains SDHW-Schematics.bmp, which we will use as the main picture. That picture was created from a screen shot of the TRNSYS Studio after hiding the output layer to simplify the schema.

We will include that picture in the first group. Just add the following line of code before the closing command for the group, after the "*|*A Simple TRNSED Demo" line:

```
*|<img src="SDHW-Schematics.bmp">
```

The syntax used to add pictures is very close to the HTML syntax (with the additional "*|" code at the beginning of the line.

1.7.4.2. Links

You will notice that the picture includes a "question mark icon" that invites user to click for help. TRNSED allows you to associate actions with pictures. In this simple example, we will launch a help file (HTML document) when the user clicks anywhere in the picture.

> TRNSED allows to define "hot spots" or clickable areas within a picture, each of them being associated with a different action. In our example, we could have the question mark open an HTML document with global info on the simulation, and the solar collector open a PDF with solar collector catalogue data. Links between tabs, which will be described in this section, can also be associated with hot spots in pictures.

To associate a link to a file or program with a picture, just add a "href=…" instruction to the "img" tag:

```
*|<img src="SDHW-Schematics.bmp" href="SDHW-Help.html">
```

You can also display a hint for users when they move the mouse over the picture by adding a "hint" instruction:

```
*|<img src="SDHW-Schematics.bmp" href="SDHW-Help.html" hint="Click for help">
```

1.7.4.3. Multiple tabs

Now that we have a picture on the main page, it might be a good idea to distribute the components in different tabs so that the user does not have to scroll down too much to make changes to the configuration.

Tabs are similar to groups in that they require an opening and a closing tag, and they have a name:

```
 *|<TabWindow name="Solar Collector">
…
```

```
All TRNSED commands that should appear in the "solar collector" tab
…
*|</TabWindow>
```

In this simple example, we can include each component (each group we created before) in a separate tab. For the "Simulation parameters group, we have:

```
*|<TabWindow name="Simulation parameters">
*|[SIMPAR|Simulation Parameters
EQUATIONS 5
I_MONTH=3624
…
ASSIGN "..\..\Weather\Meteonorm\Europe\CH-Zuerich-Kloten-66700.tm2" 36
*|? Weather data file |1000
*|]
*|</TabWindow>
```

> Note: In this case, each tab window has only one group. In general, you can include several groups in a tab. However, each group can only be part of one tab: the same group cannot span across several tab windows.

The user will be able to navigate between tabs by clicking on their name. Tabs can also be linked by pictures. We will add a "Back" link to the main tab from all other ones. In order to do this, we add a group to the "Simulation Parameters" Tab window:

```
*|<TabWindow name="Simulation parameters">

*|[SIMPAR|Simulation Parameters
…
*|]
*|[SIMPAR-Back|
*|<img src="Back.bmp" align="center" href="#Main" hint="back to main">
*|]

*|</TabWindow>
```

And we can do the same for all tabs. The "Solar collector" tab now looks like Figure 1–53.

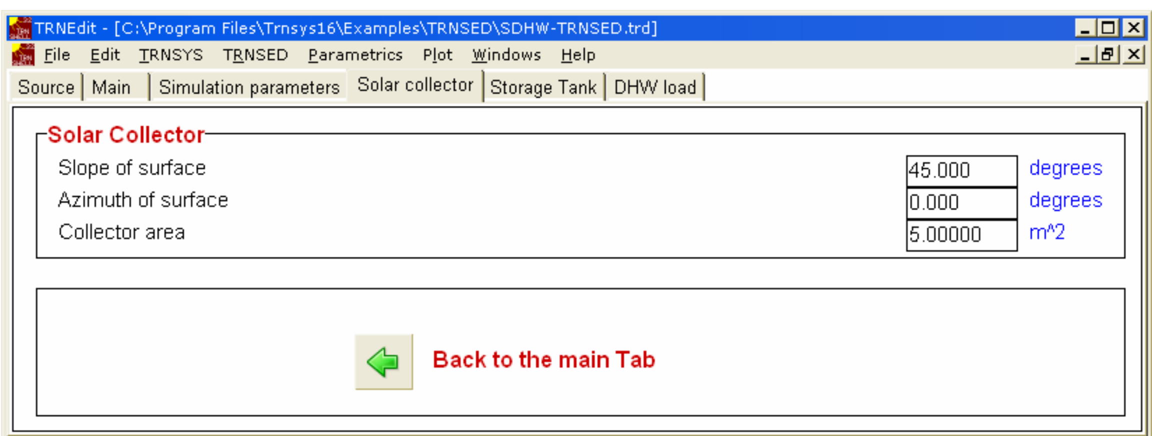

Figure 1–53: Adding a "Back" button

Please refer to Volume 7 of the documentation for more information about other TRNSED commands.

1.7.5. Creating the redistributable application

1.7.5.1. Preparing the file

You cannot be sure where users will install your application and you cannot rely on TRNSYS files to be present on their machine. In order to make sure that users will be able to use your application, you need to go through the input file and search for all absolute path references that are likely to fail on a different machine. Search for "C:\Program Files\Trnsys16" for example.

In our case, we need to change two lines to remove the path to the .exe directory and copy those two files to the Examples\SDHW-TRNSED directory: month1.dat and length.dat. Replace :

```
*|<Month of the simulation |C:\Program Files\Trnsys16\Exe\Month1.dat…
…
*|<Length of Simulation  |C:\Program Files\Trnsys16\Exe\Length.dat …
```

with:

```
*|<Month of the simulation |Month1.dat…
…
*|<Length of Simulation  |Length.dat …
```

and copy Month1.dat and Length.dat from Exe\ to Examples\SDHW-TRNSED\

The weather file uses a relative path but it is probably better to save a copy of the different files you want to send to the users in the current folder: in our case we will just copy "CH-Zuerich-Kloten-66700.tm2" from Weather\Meteonorm\Europe\ to Examples\SDHW-TRNSED\

```
ASSIGN "..\..\Weather\Meteonorm\Europe\CH-Zuerich-Kloten-66700.tm2" 36
```

With

```
ASSIGN "CH-Zuerich-Kloten-66700.tm2" 36
```

1.7.5.2. Creating the distributable

Go to "TRNSED", then "Create Distributable" and enter the following information (see Figure 1–54):

- .Exe name: Name of the program that users will have to run (SDHE-TRNSED here)
- Title: Any title you wish to use. It will be displayed on the welcome dialog box.
- Author: will be displayed at startup too.
- Destination directory: Location where you want to place all files required to run the .exe
- Include the following files: Select the .trd file(s) that should be included. Make sure you delete lines that list files which you do not want to include.

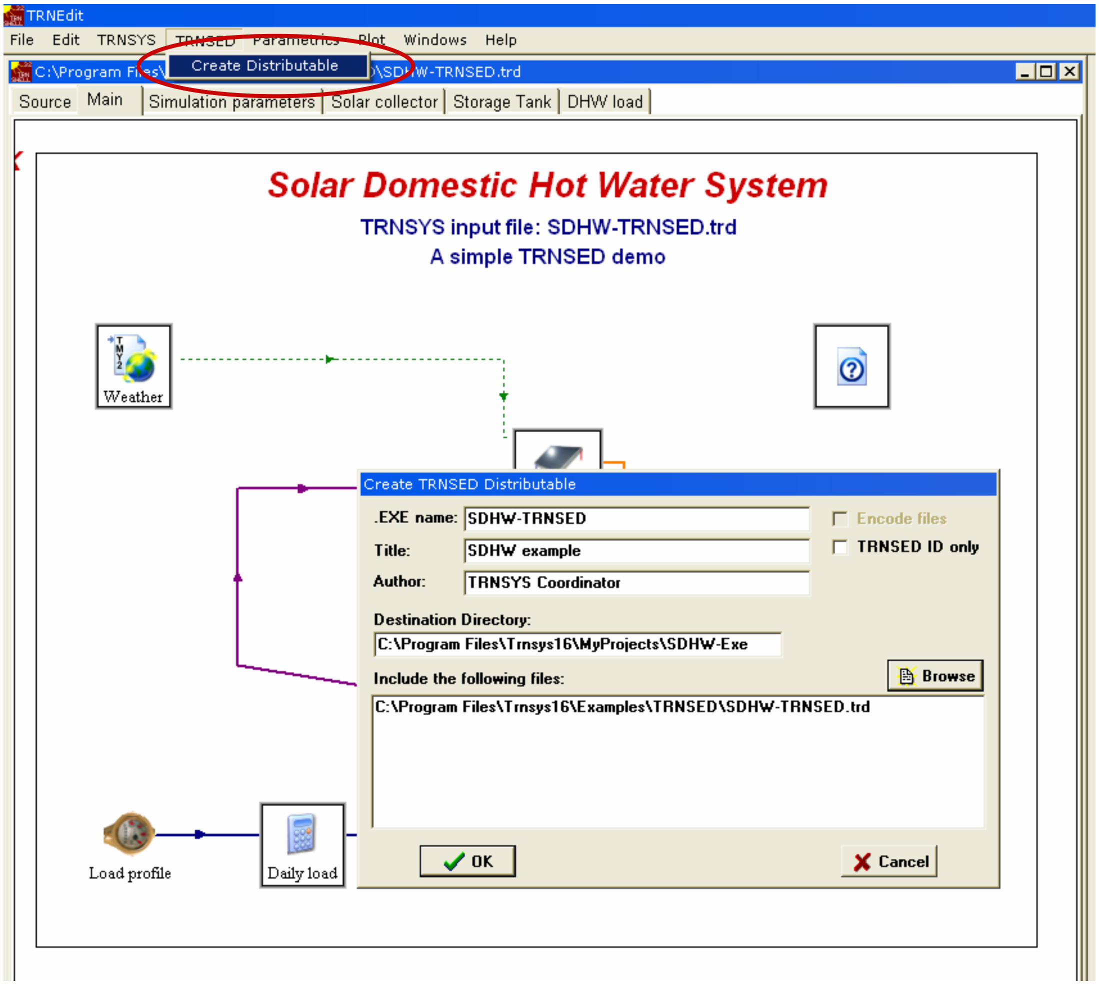

Figure 1–54: Creating the TRNSED distributable

Click OK. TRNEdit copies all the required TRNSYS files to the destination directory (MyProjects\SDHW-Exe here).

> Note: TRNEdit will copy the EXE programs, TRNDll and User libraries found in the UserLib folder. It also copies other required TRNSYS files but not additional input files that your simulation may use. For that reason, it is recommended to keep all those files in one directory or in subdirectories under the .trd file, so that you can easily copy all required files to the destination folder. You should also make sure that you use relative paths to access those files as you cannot be sure where users will install the application

In our case, the simulation uses 2 pictures: SDHW-Schematics.bmp and Back.bmp. We need to copy them from "C:\Program Files\Trnsys16\Examples\TRNSED" to "C:\Program Files\Trnsys16\MyProjects\SDHW-Exe". We must do the same with month1.dat, length.dat, the help file SDHW-Help.html, and the weather file, CH-Zuerich-Kloten-66700.tm2.

After doing this, you can launch SDHW-TRNSED.exe (or the name you gave to the exe).It is recommended that you test the distributable after moving the destination directory outside "C:\Program Files" and temporarily renaming your "C:\Program Files\Trnsys16" folder, in order to make sure that it is not using any file from your TRNSYS installation. Testing the application on another machine is also recommended.

1.8. Running TRNSYS 15 projects

1.8.1. Introduction

> Note: This section lists the basic steps that are necessary to run existing TRNSYS 15 projects in TRNSYS 16. It is assumed that those projects only use Standard TRNSYS components. If your TRNSYS 15 projects use non-standard components, you will first need to add those components after updating them, according to the instructions given in the Programmer's Guide.

There are 3 options to run TRNSYS 15 projects in TRNSYS 16:

- You can open an existing IISiBat project with the Simulation Studio. The project will be updated to TRNSYS 16, i.e. necessary adjustments will be made to the parameters and to the simulation control statements. The component proformas will be replaced with their new version.
- You can import an existing input (deck) file in the Simulation Studio. A TRNSYS 16 project (.tpf) will be generated using the new component proformas and the parameters and simulation control statements will be adapted to TRNSYS 16.
- You can run an existing input (deck) file as-is, and TRNSYS 16 will make the necessary adjustments to the component parameters and to the simulation control statements.

This section will go through the process step-by-step for 2 examples that can be found in %TRNSYS16%\Examples\Trnsys15: "Begin" and "Building". Those two examples are standard TRNSYS 15 examples that have only been adapted to remove the absolute path references.

For all TRNSYS 15 examples, different sets of files are provided in order to illustrate the various options. Those sets of files are placed in 4 subfolders:

- Original: Those are the original TRNSYS 15 project files, for reference. The only modification compared to the files distributed with TRNSYS 15 is that absolute paths have been removed
- LegacyMode: Files obtained by running the TRNSYS 15 input file (deck) as-is from TRNEdit
- StudioUpdate: Files obtained after opening the IISiBat project (.tpf) in the TRNSYS 16 Simulation Studio
- StudioImport: Files obtained after importing the TRNSYS 15 input file (deck) in the TRNSYS 16 Simulation Studio.

1.8.2. The TRNSYS 15 "Begin" example

1.8.2.1. Running the Version 15 deck file

- Launch TRNEdit
- Open "%TRNSYS16%\Examples\Trnsys15\Begin\LegacyMode\Begin-V15.dck". You will need to change the default filter from "TRNSED files" to "TRNSYS files")

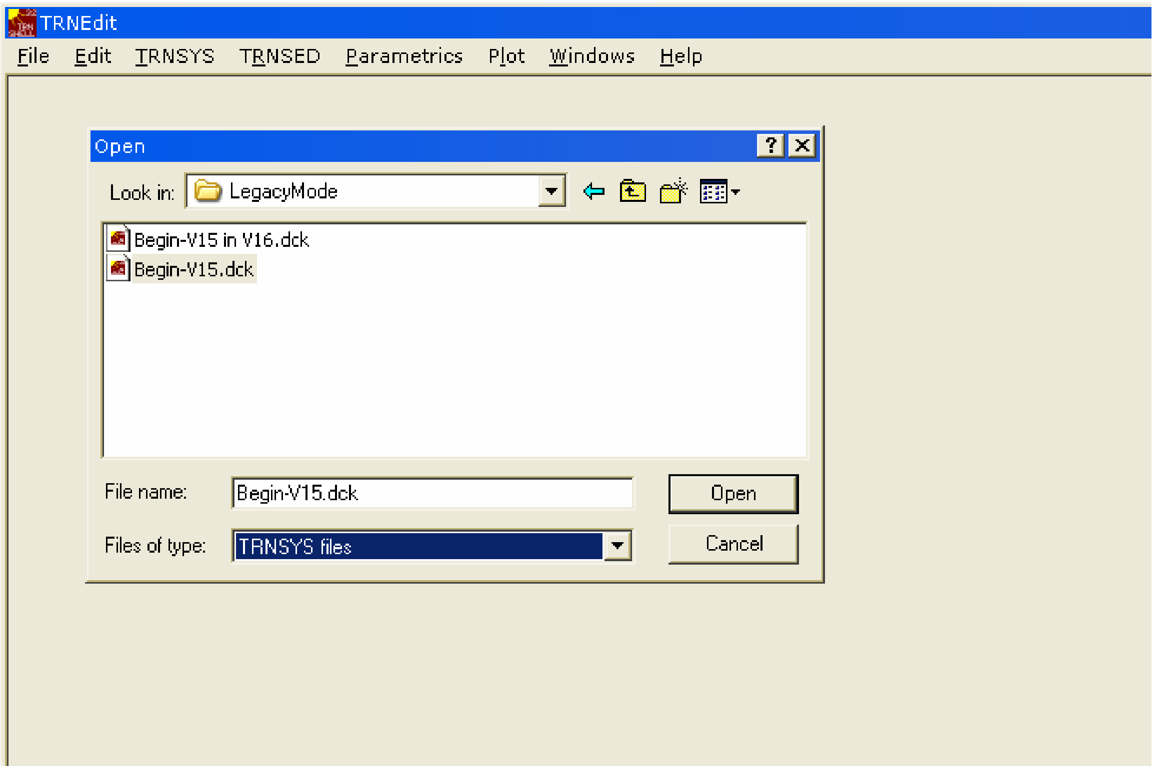

Figure 1–55: Opening a TRNSYS 15 file in TRNEdit

- Save the file as "Begin-V15 in V16.dck" (yes you can use spaces!)
- Run the file (F8). It should run without problems.
- Open the Log file created by TRNSYS (Windows / Log), see Figure 1–56
- You will see a few notices and warnings from TRNSYS. The most important ones are listed hereunder:
  - A warning telling you that a Version 15 file was recognized
  - A warning informing you that the "ASSIGN" statement for the list file has been ignored. In TRNSYS 16, the list file automatically takes the deck filename. A log file (at which you are currently looking now) is also created. The log file has the same filename as well, with a .log extension, and includes all messages from TRNSYS: notices, warnings and errors.
  - A warning informing you that the start time has been changed (in this case one hour has been removed). In TRNSYS 16, the start time is an initial time step: Types are only called

once at that time step and will output their initial values. This is a significant change from TRNSYS 15 and before, where the start time was the end of the first time step.

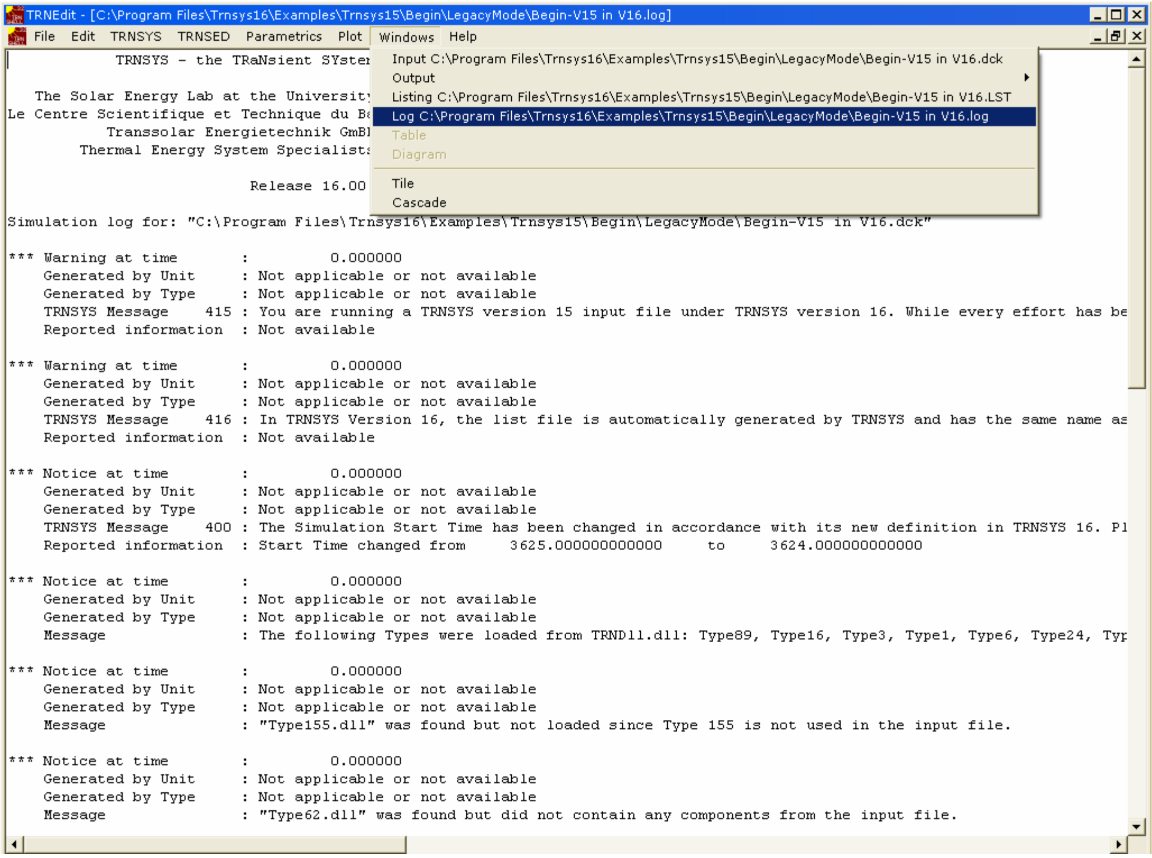

Figure 1–56: Viewing the Log file in TRNEdit

1.8.2.2. Updating the IISiBat project

- Launch the Simulation Studio
- Open "%TRNSYS16\Examples\Trnsys15\Begin\StudioUpdate\Begin-V15.tpf". The Studio will detect that you are opening a Version 15 project and will update it. The main changes to your files are listed in a dialog box that is displayed when you open the project (see Figure 1–57).
- The Simulation Studio will replace all proformas that have been updated by their new version, changing the parameters that need to be changed. This is especially true for data readers, which need to be adapted to the new convention for the simulation start time (The Studio will also change the simulation start time from the end of the first time step to the real beginning of the simulation).

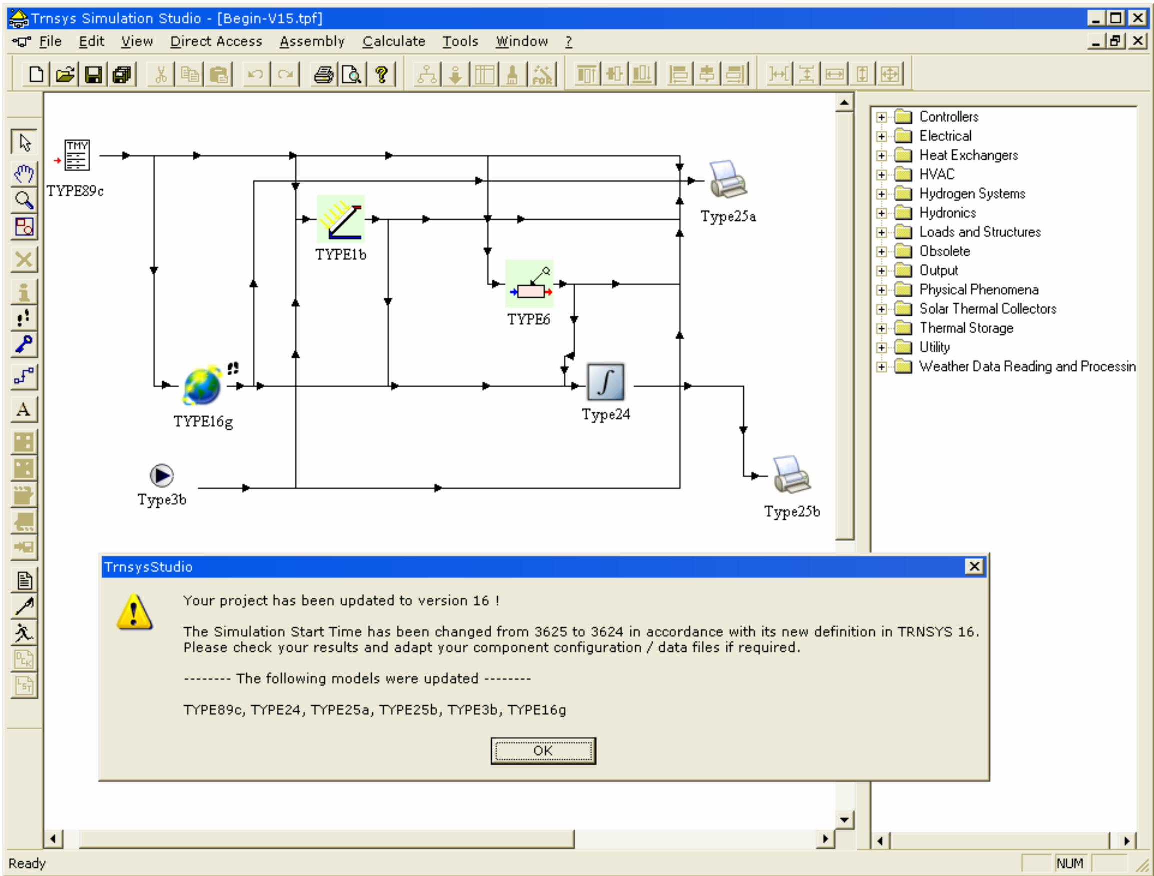

Figure 1–57: Updating an IlsiBat project in the Simulation Studio

You will notice that some icons are not updated. The reason is that the Studio does not update proformas that have not changed between TRNSYS 15 and TRNSYS 16. You can force an update if you right-click on a component and "Replace" it by itself, as illustrated in Figure 1–58.

You can save the project as "Begin-V15 updated to V16.tpf", and adapt the deck file name if you wish (e.g. to "Begin-V15 updated to V16.dck") in the control cards. If you run the file, you can check in the error manager (LST icon) that the warnings about running a Version 15 deck file are not present, since the updated project now generates a Version 16 deck file.

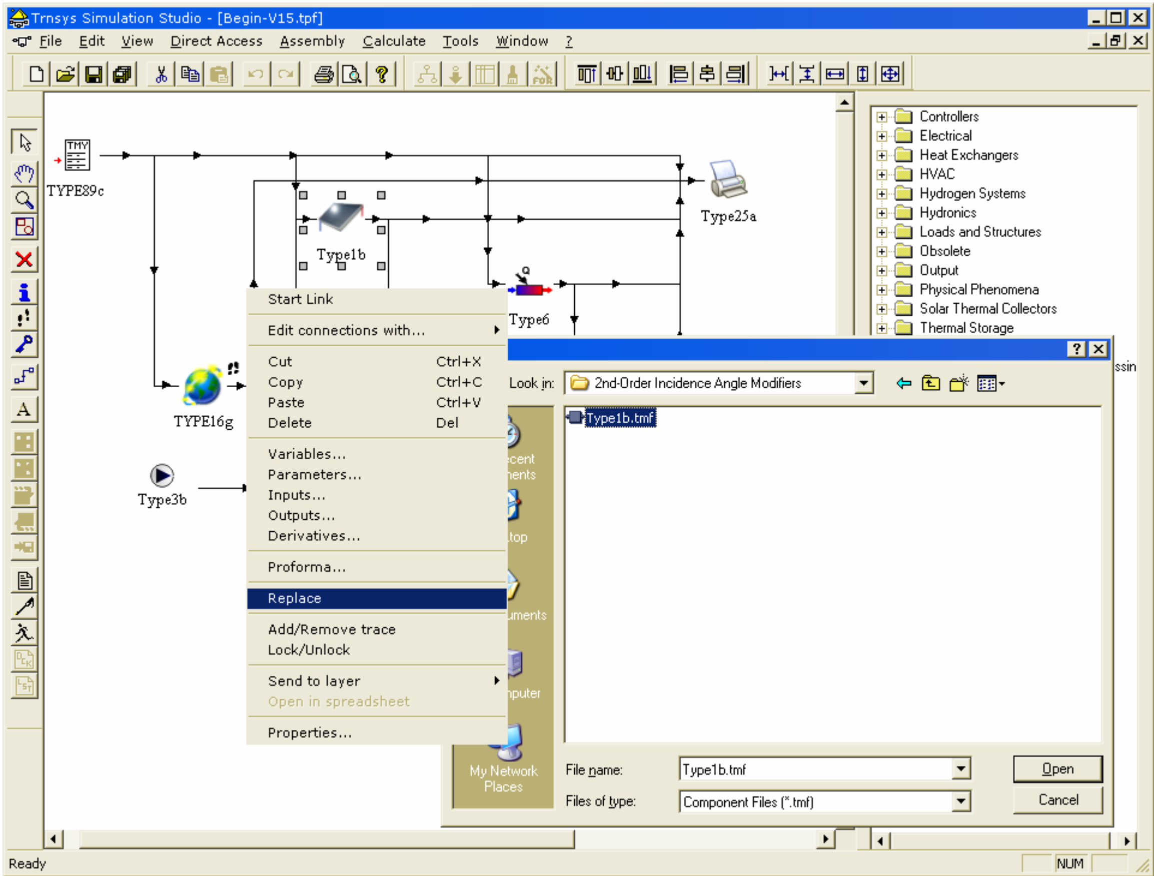

Figure 1–58: Replacing components that have not been updated

1.8.2.3. Importing the Version 15 deck file in the Studio

- Launch the Simulation Studio
- Go to File / Import TRNSYS input file
- Select "%TRNSYS16\Examples\Trnsys15\Begin\StudioImport\Begin-V15.dck"
- The Simulation Studio opens the deck file and creates a project. It displays the same messages as when you opened the Version 15 .tpf, since the same changes are made to the components.
- All icons and proformas are refreshed since the deck file only had information about the component types, their configuration and their connections.
- By default, the project gets the name of the deck file and appends "_imported" to it.
- You can run the file and check in the error manager (LST icon) that the import was fully successful.

1.8.3. The TRNSYS 15 "Building" example

1.8.3.1. Running the Version 15 deck file

- Launch TRNEdit
- Open "%TRNSYS16%\Examples\Trnsys15\Building\LegacyMode\Building-V15.dck".
- Run the file (F8). You get an error message that indicates that the gentrn.dll version used to generate the building description file (.bld) is not the one that was expected ().

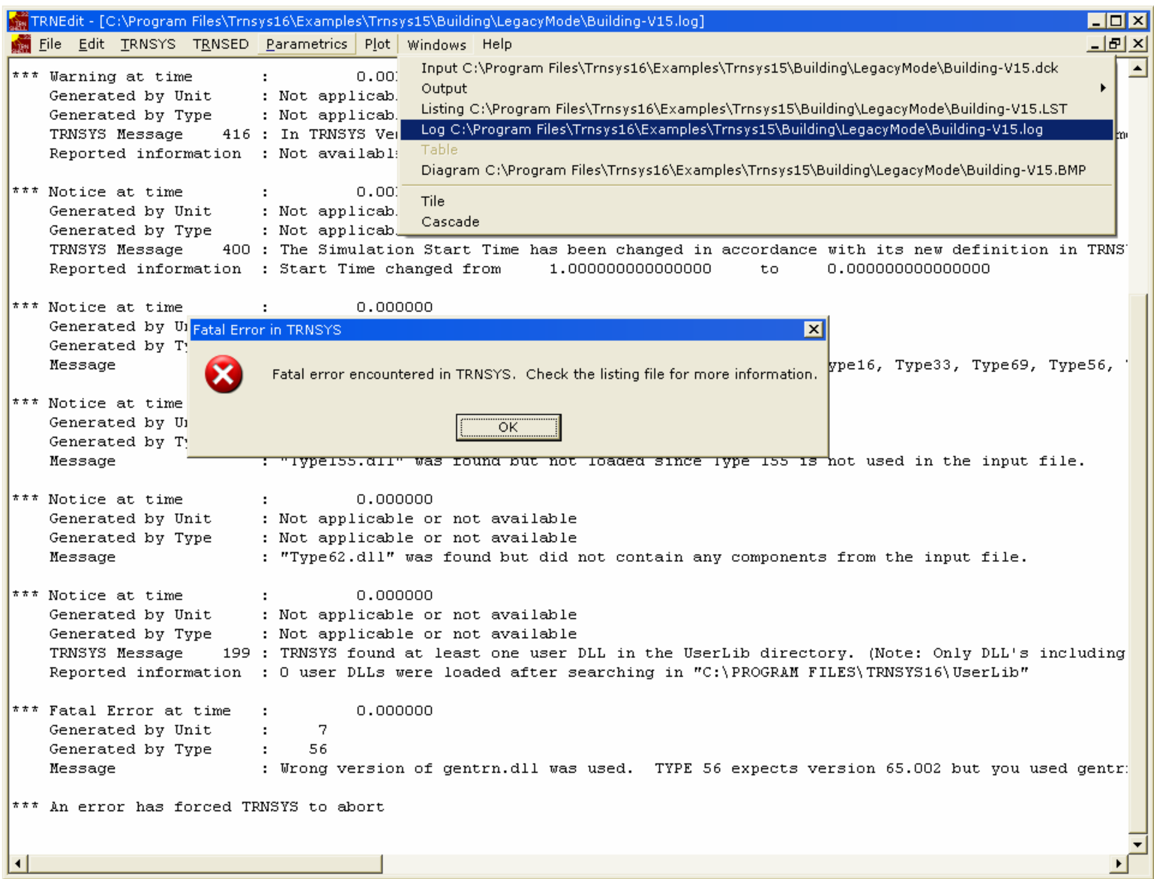

Figure 1–59: Running a TRNSYS 15 building project

The error is normal since the .bld and .trn files have been generated with TRNSYS 15. To solve that problem, you just need to open the "Bulding.bui" file in TRNBuild and save it:

- Launch TRNBuild
- Open "%TRNSYS16%\Examples\Trnsys15\Building\LegacyMode\Building-V15.bui".
- You get a warning that some windows are not present in the Win-ID pool (see Figure 1–60).

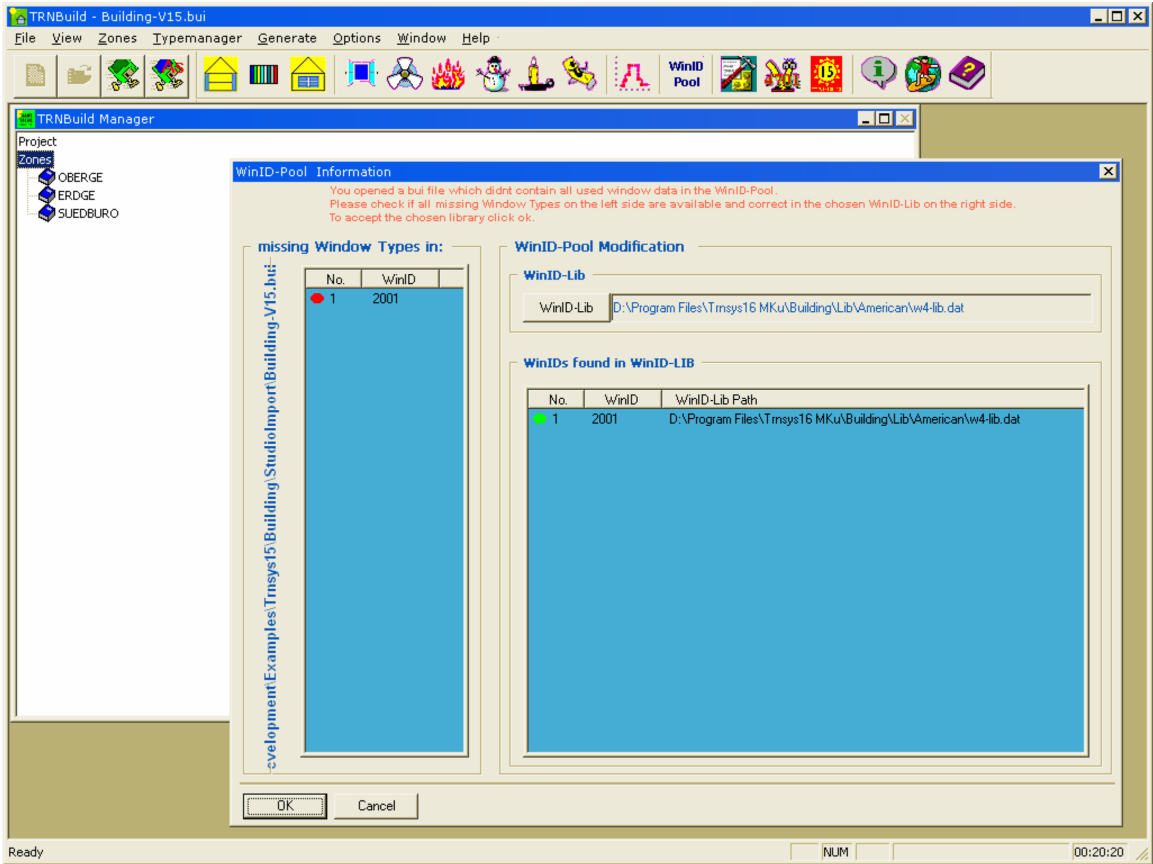

Figure 1–60: Updating a TRNSYS 15 Building file in TRNBuild

This is normal since the Win-ID pool has been introduced in TRNSYS 16. The Window-ID pool is a set of windows for which data is included in the Building file (.bui) itself, so that it is not necessary to read the glazing library (w4-lib.dat) at run-time (refer to the TRNBuild manual for more details).

- Just click OK to close the warning window
- Save the file, which will re-generate the .bld, .trn and .inf files
- After regenerating the building files, go back to TRNEdit and run the simulation again (F8). It should run without any problems now

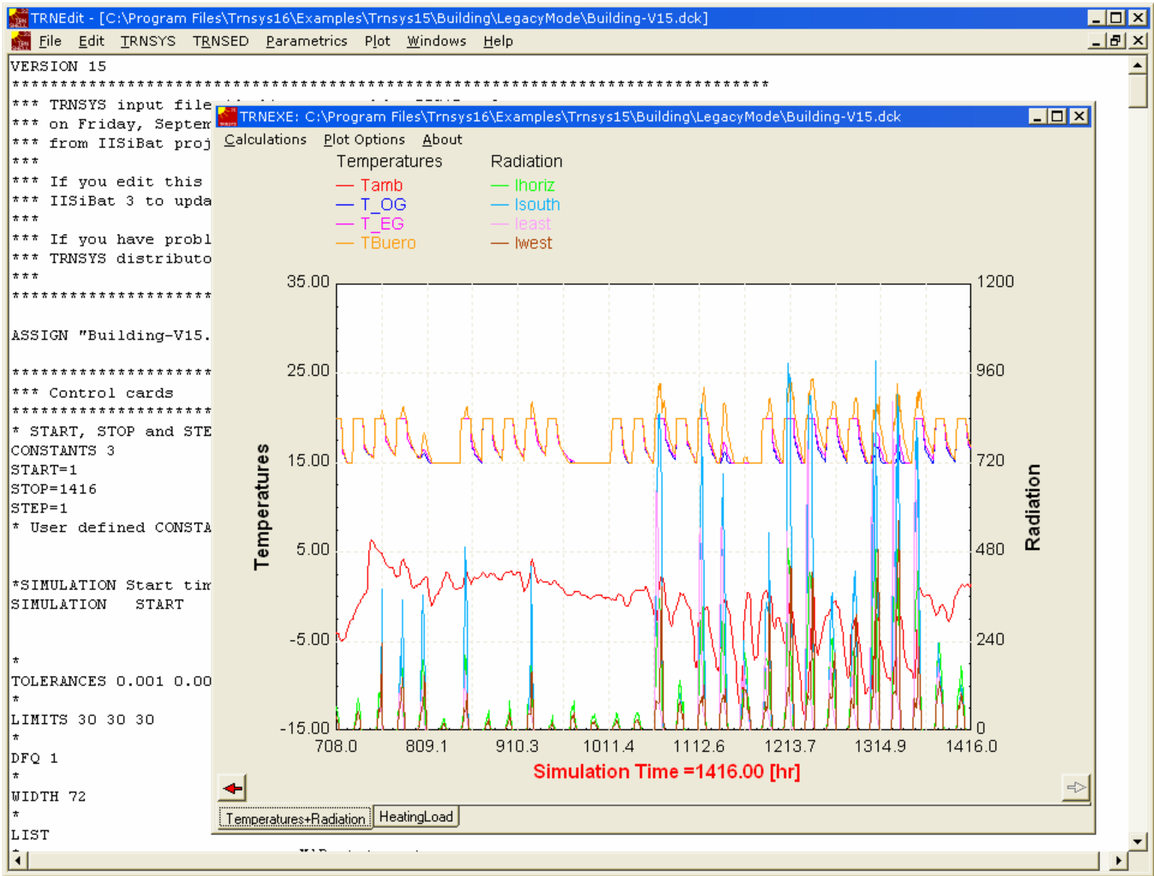

Figure 1–61: Running a Version 15 Building project after updating the building files

1.8.3.2. *Updating the IISiBat project*

- Launch the Simulation Studio
- Open "%TRNSYS16\Examples\Trnsys15\Building\StudioUpdate\Building-V15.tpf". The Studio will detect that you are opening a Version 15 project and will update it (see Figure 1–62).
- You will first see a warning that the building file needs to be updated. Click OK. The Studio launches TRNbuild to update the building file, and you see the warning about the Win-ID pool shown in Figure 1–60. Just click OK to close the window.
- The next dialog box will inform you about the main changes to your project.
- The Simulation Studio will replace all proformas that have been updated by their new version, changing the parameters that need to be changed. This is especially true for data readers, which need to be adapted to the new convention for the simulation start time (The Studio will also change the simulation start time from the end of the first time step to the real beginning of the simulation).

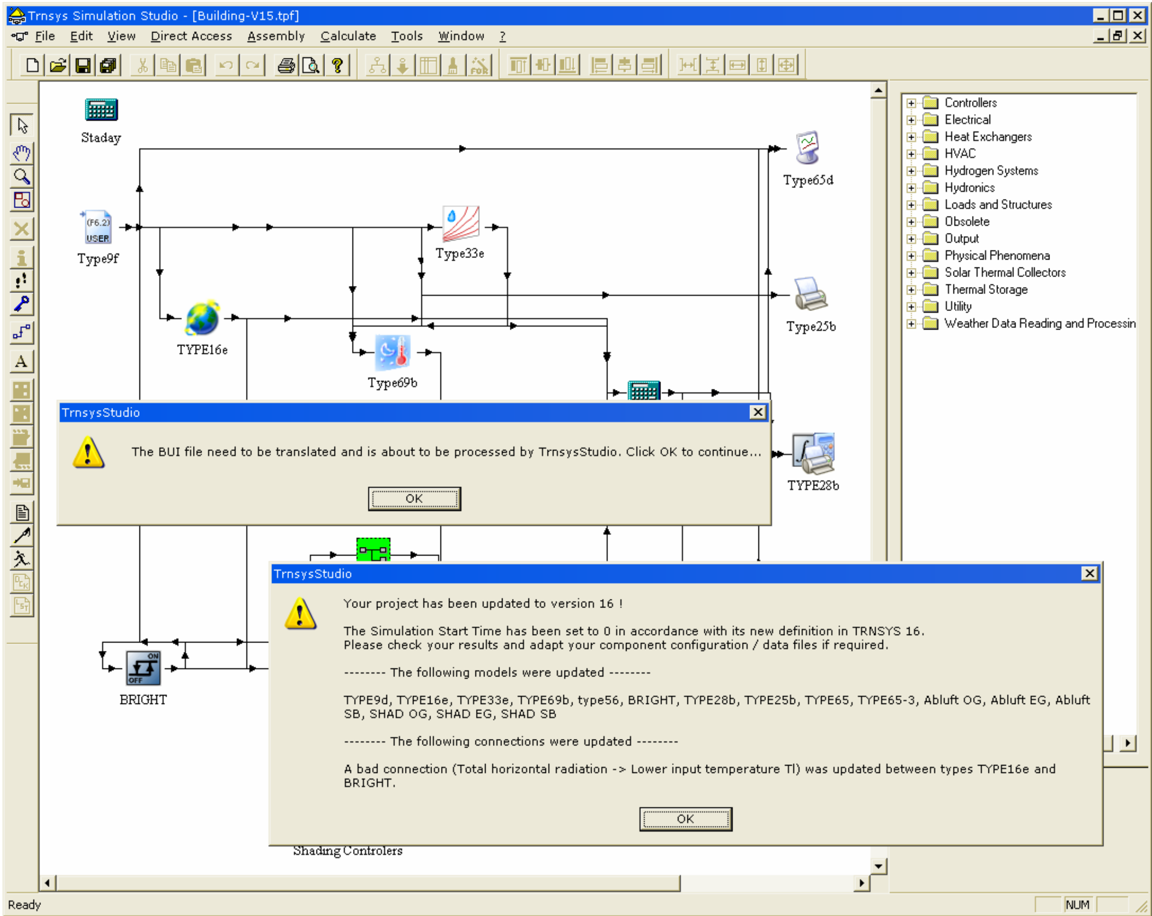

Figure 1–62: Updating a building project in the Simulation Studio

You will notice that some icons are not updated. The reason is that the Studio does not update proformas that have not changed between TRNSYS 15 and TRNSYS 16. You can force an update if you right-click on a component and "Replace" it by itself, as illustrated before in Figure 1–58.

You can save the project as "Building-V15 updated to V16.tpf", and adapt the deck file name if you wish (e.g. to "Building-V15 updated to V16.dck") in the control cards.

Run the file (F8). You will notice that TRNSYS automatically calls TRNBuild to regenerate the building files before running the simulation. This is a standard procedure in TRNSYS 16. With that automatic call and the inclusion of window data in the .bui file make it unnecessary to have any other file than the .bui before running a simulation.

If you run the file, you can check in the error manager (LST icon) that the warnings about running a Version 15 deck file are not present, since the updated project now generates a Version 16 deck file.

You will notice that the warning shown in pool shown in Figure 1–60 is displayed at each run. This is because the automatic call to TRNBuild when the Studio updates a project or when you run a simulation does not actually save the .bui file. It just regenerates the .bld, .trn, .inf files. To suppress the warning, you need to open the building file in TRNBuild and manually Save the file.

1.8.3.3. Importing the Version 15 deck file in the Studio

- Launch the Simulation Studio
- Go to File / Import TRNSYS input file
- Select "%TRNSYS16\Examples\Trnsys15\Building\StudioImport\Building-V15.dck"
- You will first see a warning that the building file needs to be updated. Click OK. The Studio launches TRNbuild to update the building file, and you see the warning about the Win-ID pool shown in Figure 1–60. Just click OK to close the window.
- You are then informed that warnings or errors were generated during the import (see Figure 1–63). The warning is not important as it actually informs you that there was a bad connection between a solar radiation processor and a controller that is supposed to have temperatures as inputs.
- The next dialog box will inform you about the main changes to your project. It displays the same messages as when you opened the Version 15 .tpf, since the same changes are made to the components.

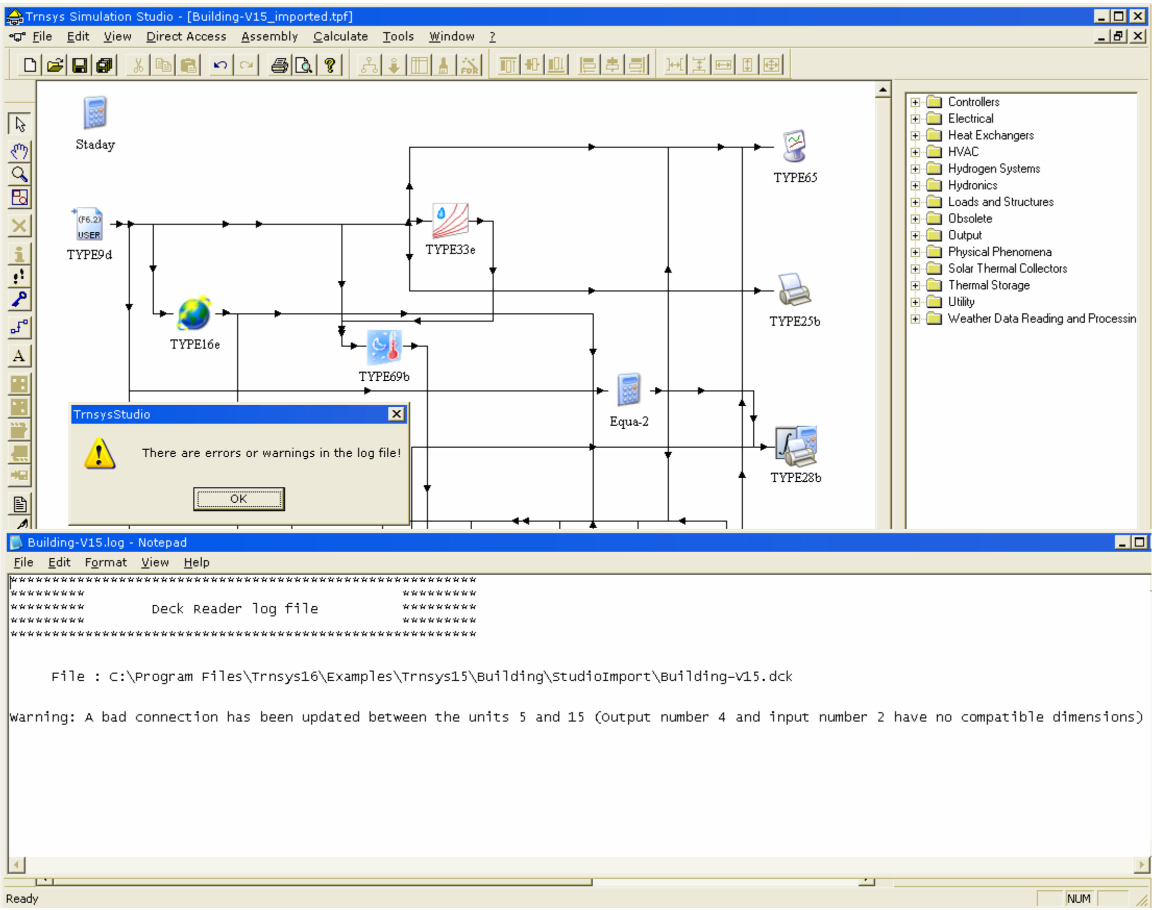

Figure 1–63: Importing a Version 15 building project in the Simulation Studio

All icons and proformas are refreshed since the deck file only had information about the component types, their configuration and their connections. By default, the project gets the name of the deck file and appends "_imported" to it.

You can run the file and check in the error manager (LST icon) that the import was fully successful.

The same remark applies as when you updated the IIsiBat project. You will get warnings during each automatic call in TRNBuild unless you open the building file and save it manually.

1.9. Creating a new component

This section will briefly illustrate how to generate new components using the Simulation Studio. Please refer to Volume 8 (Programmer's guide) for detailed instructions on creating new components, compiling them and linking them into an external DLL.

1.9.1. Create the component proforma

- Launch the Simulation Studio and go to File/New
- Select "component" (see Figure 1–64)
- In the component "General" tab, type in the component's object and its Type number. It is important to assign a Type number different from zero. The selected number should also be in the [201;300] range in order to avoid conflicts with existing libraries. Our component will add its two inputs and multiply by its only parameter, generating one output.

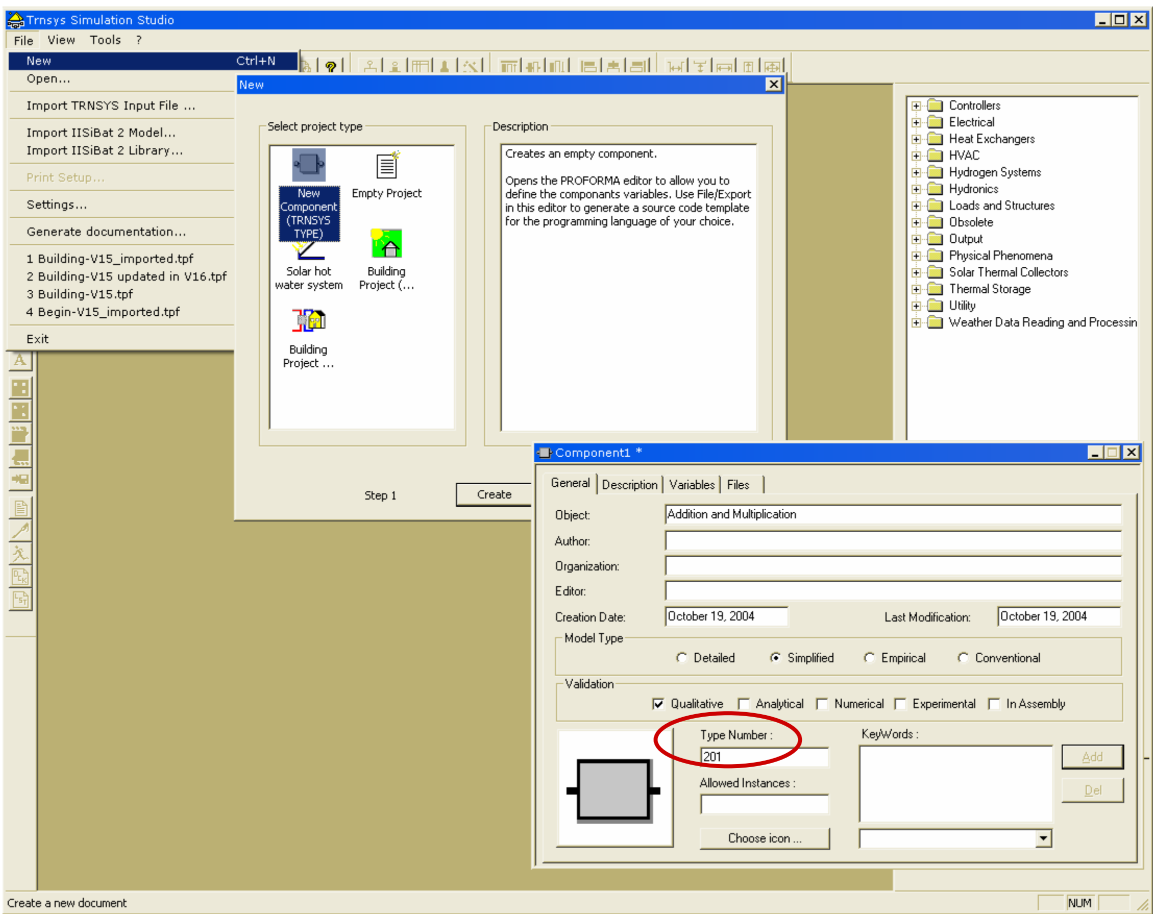

Figure 1–64: Creating a new component proforma (1)

1.9.2. Generating a Fortran code skeleton and a compiler project

You can then add variables to the proforma in the "Variables" tab. Click on "variables", select the "Parameters" tab and click on "Add". You can enter parameter information in the row that has been created or click on "Modify" to have a more detailed view (see Figure 1–65).

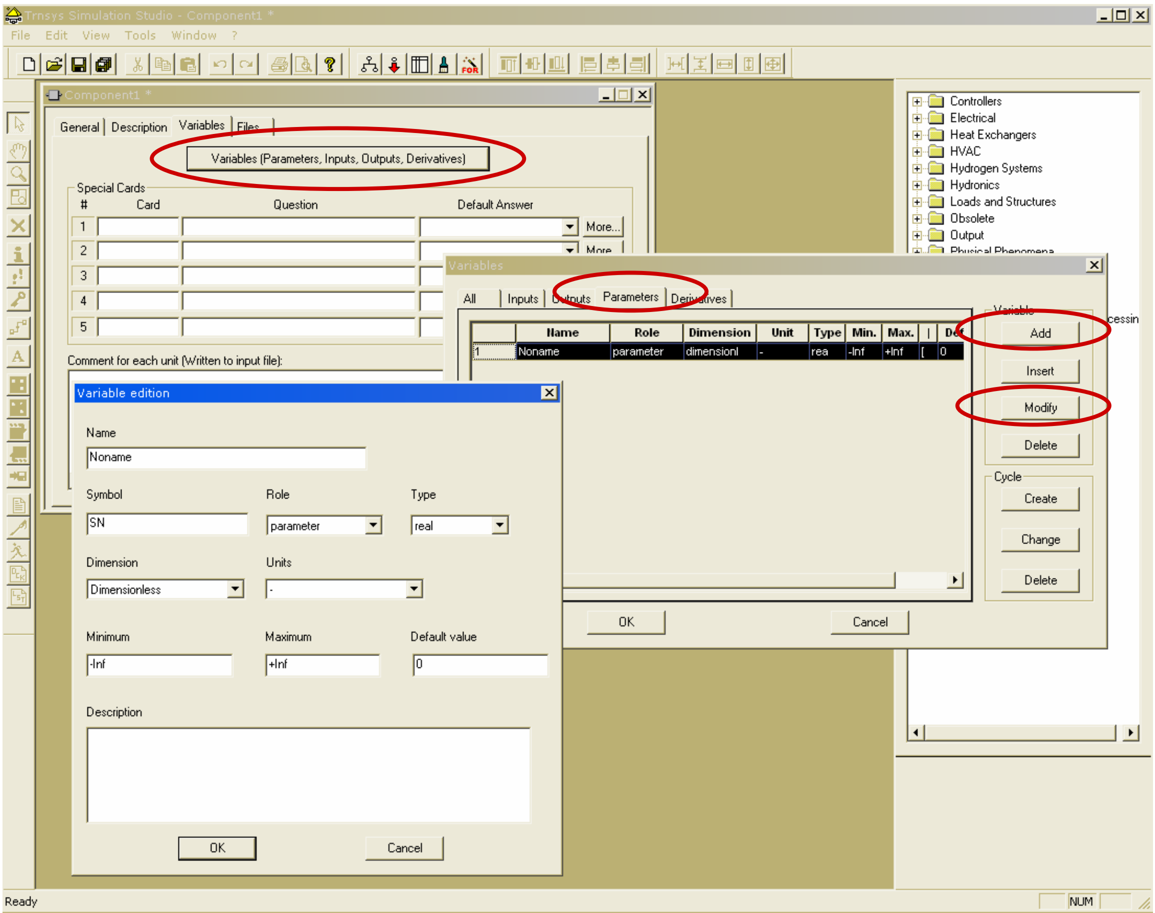

Figure 1–65: Creating a new component proforma (2)

We will call the parameter "Mult" and set its default value to 1.

You can then switch to the input tab and add two inputs, inp1 and inp2.

Finally, switch to the output tab and create one output, out1. The "Variables" dialog box is shown in Figure 1–66.

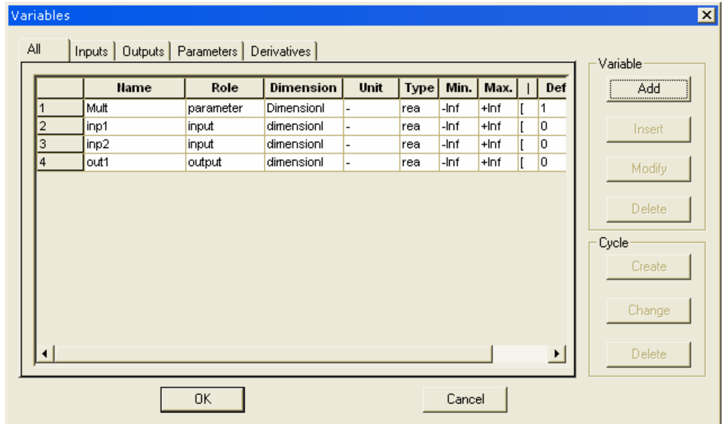

Figure 1–66: Creating a new component proforma (3)

Save the proforma. To be accessible in the Studio, the proforma needs to be in "%TRNSYS16%\Studio\Proformas". You can for example create a folder called "My components" and save your component there, as "Type200.tmf".

Go to "File/Export as/Fortran" and create a "Type200" folder anywhere on your disk (e.g. in "MyProjects"). Save the component as Type200.for there (see Figure 1–67).

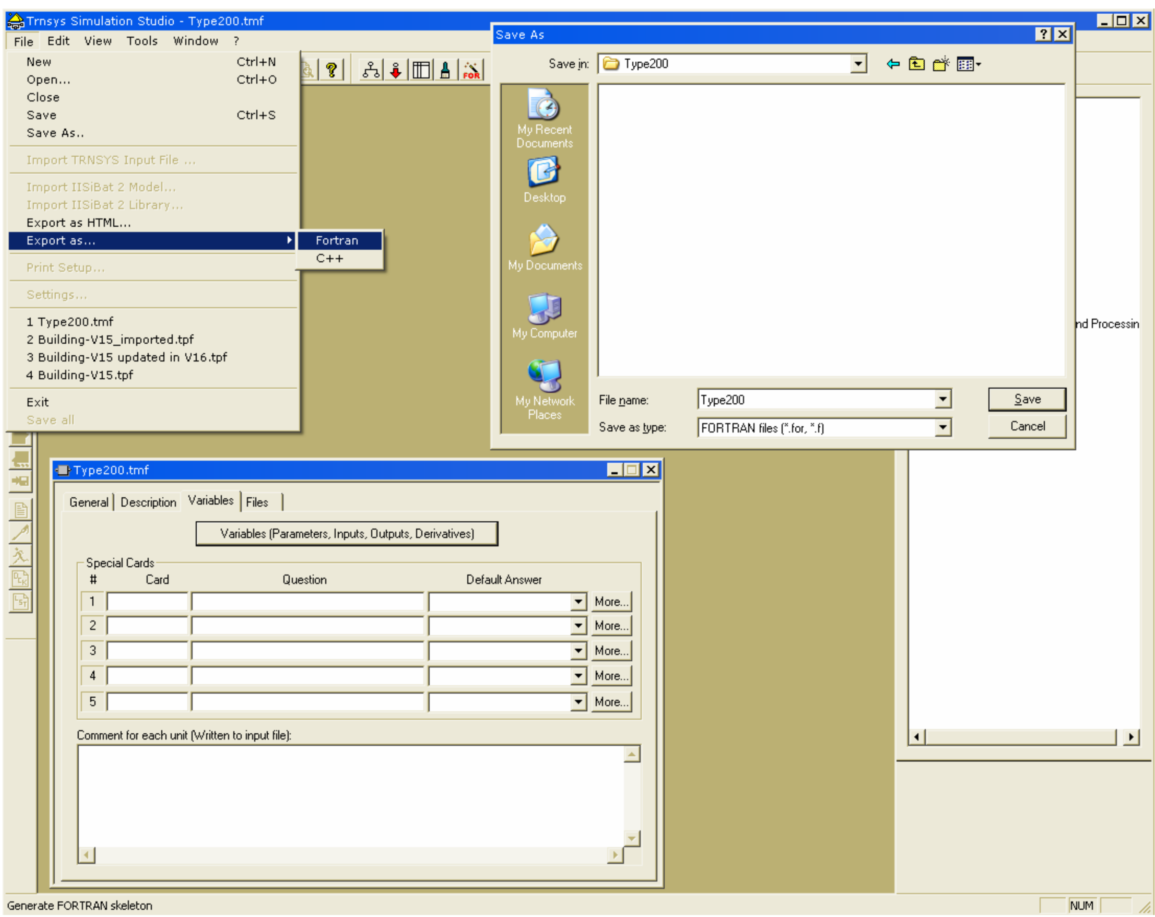

Figure 1–67: Exporting as Fortran (1)

The Studio will then do several things for you (see the dialog box in Figure 1–68):

- Create a Fortran skeleton for your Type (Type200.for)
- Generate a project that you can open in the Compaq Visual Fortran 6.6 compiler or in the Intel Visual Fortran compiler. That project includes all the settings you need in order to generate an external DLL that will be placed in the \UserLib\ReleaseDLLs or \UserLib\DebugDLLs folder, where TRNSYS will be able to load your Type.
- Open that project in the compiler that you have setup in File/Settings/Directories/Fortran environment (by default the settings are correct for CVF 6.6 if you installed it in the default location.

The following will assume that Compaq Visual Fortran 6.6 is installed on your machine.

The skeleton includes all basic TRNSYS manipulations. At the minimum, you need to replace question marks in the lines that calculate outputs from inputs (see Figure 1–68).

In our case, we will type in the following line:

```
OUT(1) = (inp1+inp2)*Mult
```

inp1 and inp2 have been declared and set to xin(1) and xin(2) respectively, and Mult has been set to par(1). Xin and par are the input and parameter arrays of a TRNSYS component.

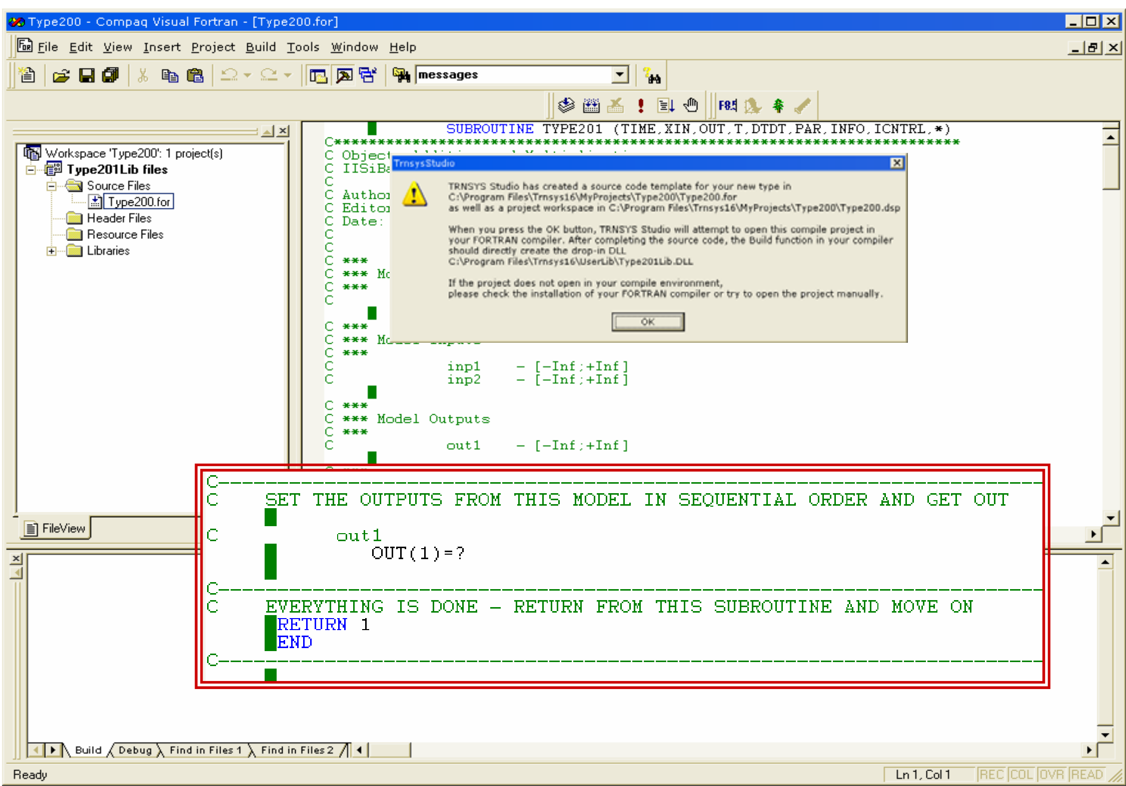

Figure 1–68: Setting the component outputs

Projects may be built in two different configurations: Debug and Release. All DLL's must be compiled in the same mode. As TRNSYS is distributed with its main library file, TRNDll.dll, built in Release mode, you must build your components in Release mode. Please go to 'Volume 8 – Programmer's Guide' for more information on how to compile your components in Debug mode.

Go to ‘Build/Set active configuration’, and select the ‘Release’ project configuration. Press F7 to build the DLL. This will generate "Type200.dll" in the ‘UserLib\ReleaseDLLs’ directory (see Figure 1–69). TRNSYS loads external libraries depending on the mode in which the main DLL was compiled: ‘UserLib\ReleaseDLLs’ and ‘UserLib\DebugDLLs’. In this case, it will use the Release directory.

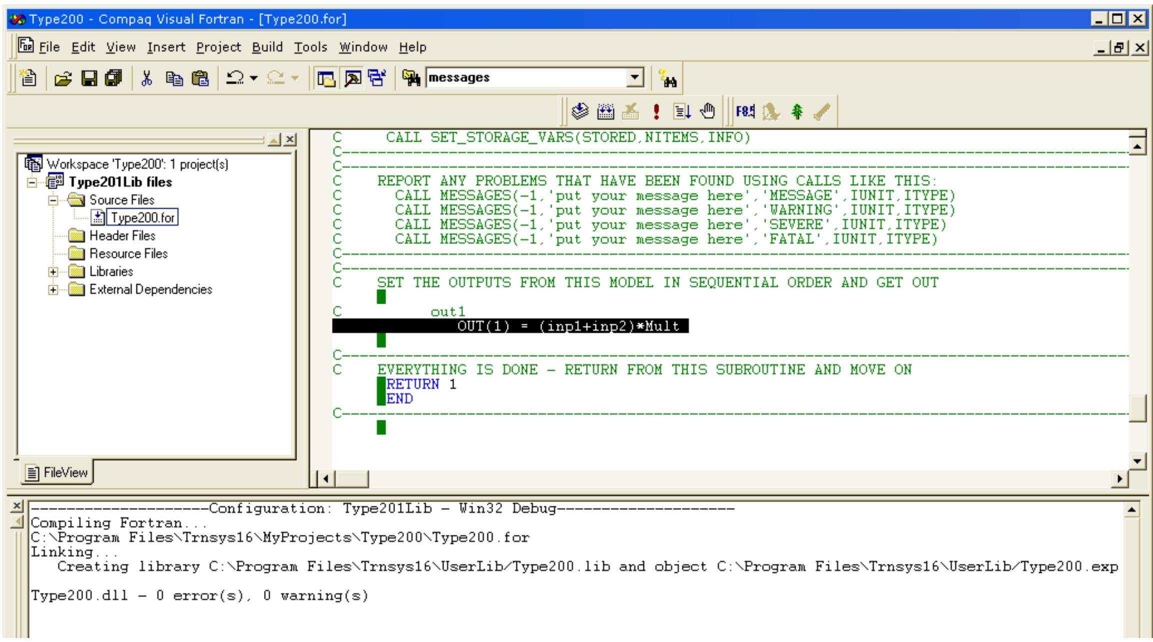

Figure 1–69: Compiling the component and building the DLL

You can now come back to the simulation Studio and create a project to use the new component. Go to "File/New/Empty Projects". In order for your newly created proforma to appear in the direct access tool (component list on the right of the screen), you need to refresh it using "Direct Access/Refresh tree" (see Figure 1–70).

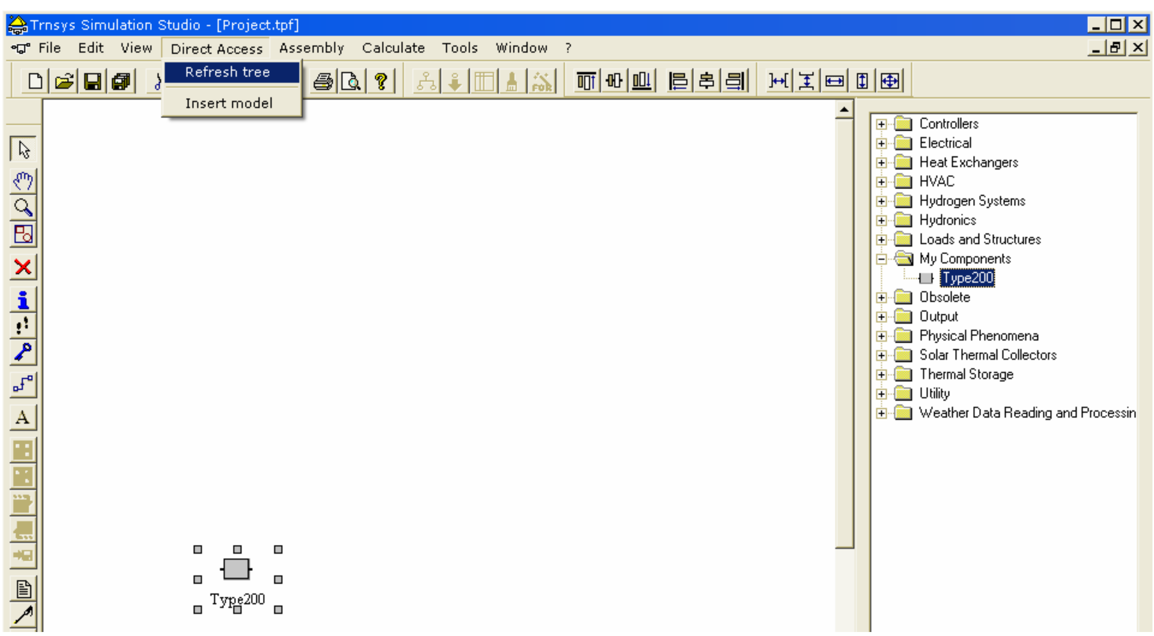

Figure 1–70: Using the new component in a project

We will just add an equation block (Assembly / Insert New Equation), in which we create two variables ("First" and "Second"). "First" is set to "time" and "Second" is set to time*2 (time is the TRNSYS simulation time, which is available to all equations).

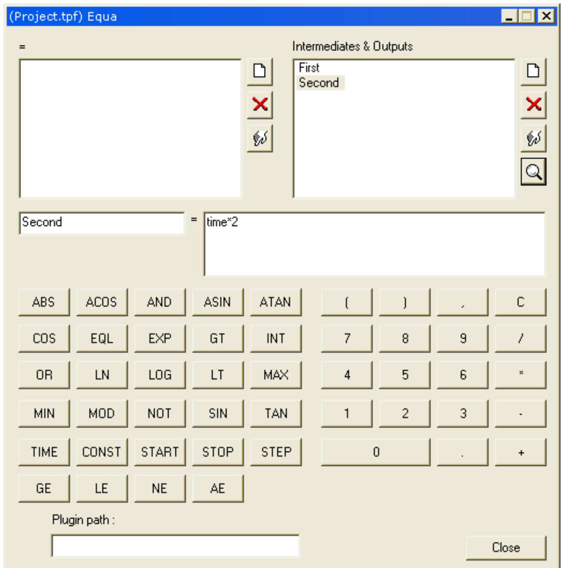

Figure 1–71: Setting the inputs using equations

We will then connect the equations to the component's inputs, set the component's only parameter to 2 and connect its output to an online plotter (Figure 1–72).

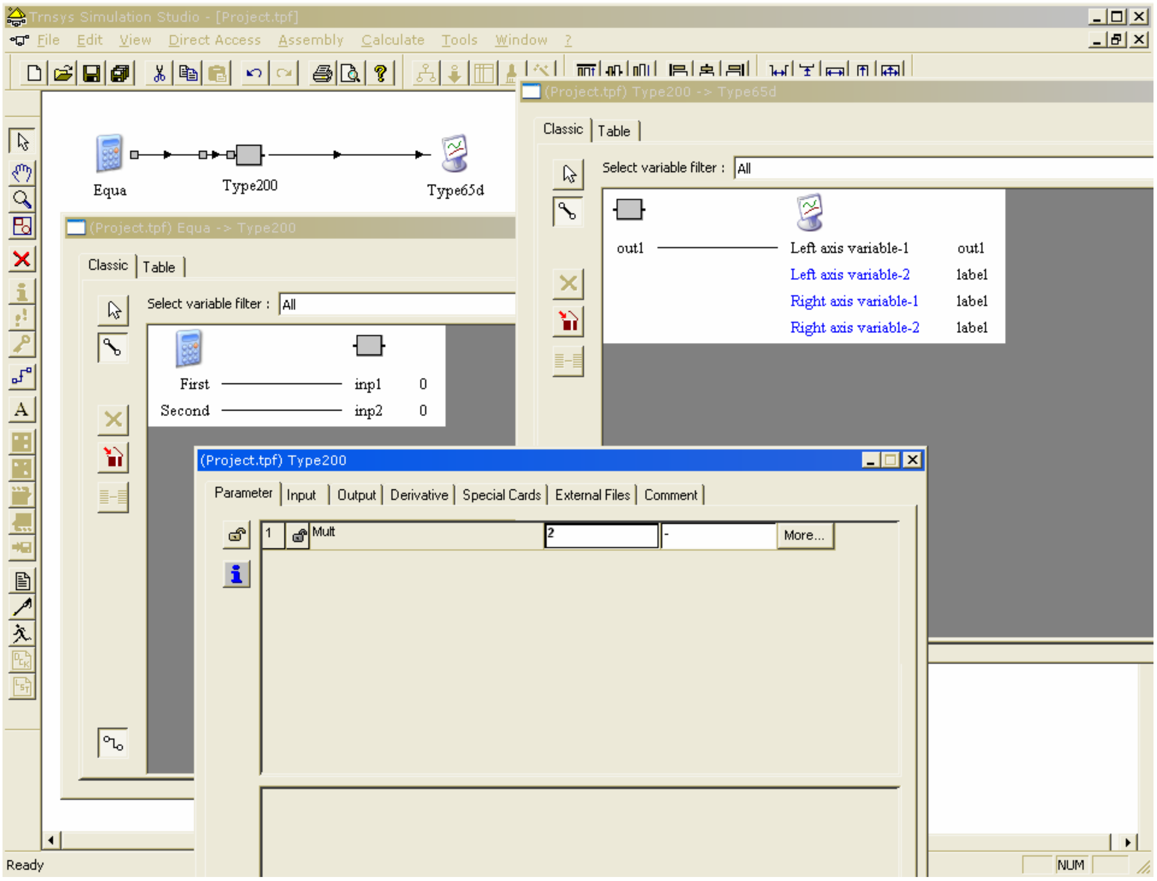

Figure 1–72: Completing the project

Run the project (F8). The output should be equal to: (time + 2*time) * 2 = 6*time. You can check the values in the online plotter by pressing CTRL-SHIFT and moving the mouse over the plot (see Figure 1–73).

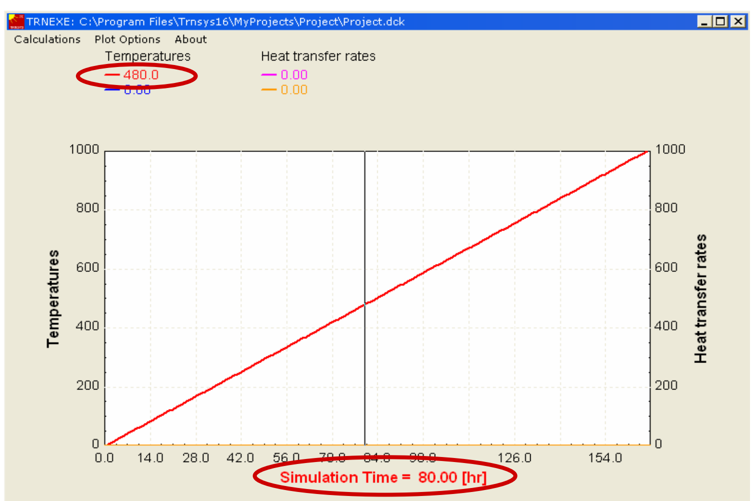

Figure 1–73: Checking the component's output

1.9.3. Using other compilers or creating Types in other programming languages

The multi-dll architecture allows you to create a Type with any Windows compiler able to generate DLL's, provided that you respect the calling conventions of TRNSYS 16.

1.9.3.1. Other Fortran compilers

If you want to generate a Fortran Type using a non-supported compiler, you can use the Fortran skeleton. You will find detailed instructions on compiler configuration in Volume 8, and you will have to adapt those settings to your compiler.

1.9.3.2. Types in C++

The Simulation Studio has another "export wizard" that will basically perform the tasks described here above to generate a C++ Type instead of a Fortran Type. Use the "Export as C/C++" command instead of "Export as Fortran". Generated projects are compatible with Microsoft Visual Studio .Net . If you use another compiler, you will have to adapt compiler settings. You can use the Trnsys.h header file that is in \Compilers\Cpp-Mvs2003\Include folder for TRNSYS function declarations.

1.9.3.3. Other languages

If you want to create a Type in another programming language, you can still use the generated C++ Type for reference (and Trnsys.h for function declarations). Refer to Volume 8 for details on TRNSYS functions and calling conventions.

1.10. Running TRNSYS in batch files or in hidden mode

1.10.1. Running TRNSYS in batch mode

You can run TRNSYS in batch mode by adding a "/n" switch to the command line, e.g. (on one line):

```
"C:\Program Files\Trnsys16\Exe\TRNExe.exe"
                   "C:\Program Files\Trnsys16\Examples\SDHW\SDHW.dck" /n
```

In TRNSYS 16, the behavior of the /n switch was changed so that it also skips the dialog box that informs you about errors in the simulation. Note that this is only true for actual TRNSYS errors, not for exceptions generated by the code such as floating point overflows, etc.

1.10.2. Running TRNSYS in Hidden mode

You can run TRNSYS in hidden batch mode by adding a "/h" switch to the command line, e.g. (on one line):

```
"C:\Program Files\Trnsys16\Exe\TRNExe.exe"
                   "C:\Program Files\Trnsys16\Examples\SDHW\SDHW.dck" /h
```

The "/h" switch implies the "/n" switch and makes TRNSYS completely invisible (you can check that it is running by launching Windows' task manager and look for a process called TRNExe).

> Please note that this switch is only applicable to simulations that do not use the online plotter or for which all online plotters are disabled (which can be done by setting parameter 9 to -1).

1.11. More TRNSYS Examples

You can learn more about TRNSYS 16 by using the examples included in the distribution. To open an example, launch the Simulation Studio, select "Open" and browse to %TRNSYS16%\Examples.

The following is a non-exhaustive list of examples. We encourage you to explore the TRNSYS examples in the Simulation Studio.

Begin

Simple example described in section 1.4.1

Calling Comis

Example using the link between TRNSYS and COMIS (Type 157). Requires COMIS 3.1 to be installed.

Calling CONTAM

Example using the link between TRNSYS and CONTAM (Type 97). CONTAM is required to edit the project files but not to run the example.

Calling EES

Example using the link between TRNSYS and EES (Type 66). Requires EES professional to be installed.

Calling Excel

Example using the link between TRNSYS and Excel (Type 62). Requires Microsoft Excel 97 or above to be installed.

Calling Matlab

Example using the link between TRNSYS and Matlab (Type 155). Requires Matlab R13 or above to be installed.

A second example ('Calling Matlab-MultipleUnits') shows how to run a project with multiple instances of Type155.

Calling_Fluent

Example using the link between TRNSYS and FLUENT (Type 101). Requires Fluent 6.0 or above to be installed.

Calculating Inputs from Outputs

The mathematical formulation of components imposes a causal relation between inputs and outputs. Sometimes, it is desired to calculate the inputs of a component based on a desired value of an output. By using a feedback controller, this example calculates the mass flow rate required to achieve a given heat transfer rate across a heat exchanger.

Controllers

This example shows the basic settings for a thermostat (Type8) and a generic differential controller (Type2).

Data Files

This folder contains various examples of data files used by some of the TRNSYS components.

Detailed Single Zone Building

This example illustrates how to use a detailed single-zone building model (Type 19) to calculate the hourly cooling and cooling loads, as well as the maximum and minimum loads.

Diesel Dispatch

This example illustrates the use of Type 120 (Diesel Electricity Generator Set) and Type 102 (DEGS Dispatch Controller) to adapt the number of running engine to a given electrical load. Both components were formerly part of the HYDROGEMS library authored by Øystein Ulleberg, IFE

Electrolyzer Controller

This example illustrates the use of Type 100 (Electrolyzer controller) in combination with Type 160 (electrolyzer).

Feedback Control

This directory includes different examples derived from the SDHW and Sunspace examples, as well as projects that use a simple zone model (Type 88) to demonstrate the use of Type 22 (Iterative Feedback controller) and Type 23 (PID Controller). Type23 is also used in a pair of examples intended to show how it can be used in cooling applications and in combined heating and cooling applications.

HVAC-building

This example shows how to simulate an HVAC system that connects to a multizone building, using the ventilation input for each zone.

Parametric runs

This directory includes a TRNSYS Studio project that is used as a basis to generate a deck file for a parametric study. To perform the parametric study, Open SDHW-ParametricRuns.dck in TRNEDit and then open the parametric table (SDHW-ParametricTable.tbl), or refer to Volume 7 for more information on this example.

Photovoltaics

This directory contains examples that show how to represent the IV curve of a photovoltaic panel using a 5-parameter model, and how to predict the performance of a system with photovoltaic panel and inverter.

Reading Data Files

This example shows how Type9 interprets the different parameters when it reads external data files. Please go to Volume 5 –Mathematical Reference for the explanation of this example.

Restaurant

This is the "Restaurant" example described in detail in the TRNBuild and Type 56 manual (Volume 6 of the TRNSYS documentation).

SDHW

This example models a simple Solar Domestic Hot Water (SDHW) System. It also illustrates the use of line styles to help understand the information and fluid flows in a simulation

Shading

This directory includes two examples of using Type68 (Shading Masks). In ShadingMasks.tpf, Type68 is used to calculate the shading effect of a far away mask on a row of solar thermal collector panels. In Combined68And34.tpf, a Type68 Shading Mask component is used upstream of a Type34 Overhang and Wingwall Shading component .

Stand-Alone Power System

This example illustrates a system where a Fuel cell (Type 173), an Electrolyzer (Type 160) and a Hydrogen storage tank (Type 163) are used to store the electricity generated by a Wind Turbine and produce electricity from Hydrogen when needed. The components of the Hydrogen system were formerly part of the HYDROGEMS library.

SunSpace

This example is inspired by BESTEST Case 960. It simulates a shoebox building with a glazed sunspace facing South and an opaque back zone behind it.

SunSpace-Shading

Calculating shading from overhangs and wing walls, as well as from far away objects requires the combination of Type34 and Type68. This example modifies the SunSpace example, taking into account shading.

TRNSED

This folder includes examples of inputs files that are used to generate stand-alone applications based on TRNSYS (TRNSED applications – see section 1.7).

TRNSED-Advanced

This directory includes a more advanced TRNSED example than the one that is described in this Getting Started guide. Refer to Volume 7 for more information on this example and on TRNSED features.

Trnsys15

This folder contains TRNSYS 15 projects that can be updated to TRNSYS 16 (see section 1.8)

Weather Data

This folder contains examples that illustrate the use of Weather data files distributed with TRNSYS, including US-TMY2 files generated from the NSRDB database, worldwide TMY2 files generated using Meteonorm ®

Wind Diesel

This example illustrates the use of Type 120 (Diesel Electricity Generator Set) and Type 102 (DEGS Dispatch Controller) to adapt the number of running engines to a given electrical load that is partly met by Wind Turbines (Type 90). Type 120 and Type 102 were formerly part of the HYDROGEMS library.

1.12. General description of the TRNSYS Manuals

Volume 1 – Getting Started: This manual explains what TRNSYS is and what programs make the TRNSYS suite. You will learn how to install TRNSYS, run examples and create simple projects.

Volume 2 - Using the Simulation Studio: This manual describes the TRNSYS Simulation Studio in detail. This visual interface is used to create and simulate projects. It can also help you create new components.

Volume 3 - Standard Component Library Overview: This manual gives an overview of the available components in the standard TRNSYS library.

Volume 4 - Component Inputs – Outputs – Parameters: This manual gives a detailed list of inputs, parameters and outputs for all components in the Standard Library.

Volume 5 - Component mathematical reference: This manual gives the mathematical description of all components available in the Standard TRNSYS library.

Volume 6 - Multizone Building (Type56 – TRNBuild): The TRNSYS multizone building (Type 56) and its visual interface (TRNBuild) are described in detail in this manual.

Volume 7 - Editing the Input File and creating TRNSED applications: This manual includes a description of the input file syntax. It also explains how to use TRNEdit to edit TRNSYS input files, create distributable applications (known as TRNSED apps.) and run parametric studies.

Volume 8 - Programmer's Guide: This user guide describes how the kernel and the components interact. It provides detailed instructions to create components and to easily update TRNSYS 15 components.

Volume 9 - Weather Data: This manual describes the weather data distributed with TRNSYS 16. More than 1000 files in more than 150 countries are available.