external sources

IBM® Rational® Rhapsody® StartUp Training

WILLERT.

LPC17xx.h

GLCD_SPI_LPC1700.c

glcd.h

Mixing Model and Code

Modeling software projects with UML makes a lot of fun. But sometimes we can not avoid, to use conventional C-Code in our UML project. Quite often, we have to take code from older projects, or we need to integrate third party software to our project. Unfortunately, this software is usually not available as a UML model.

So, we have to mix a UML model with normal code.
The easiest way is, of course, if we do not have to adjust our external files. Just add your sources, headers and libraries straightforward to a UML element. We have the opportunity to do this at several points in our model. But, using this method, your external files can not be placed in a diagram view.

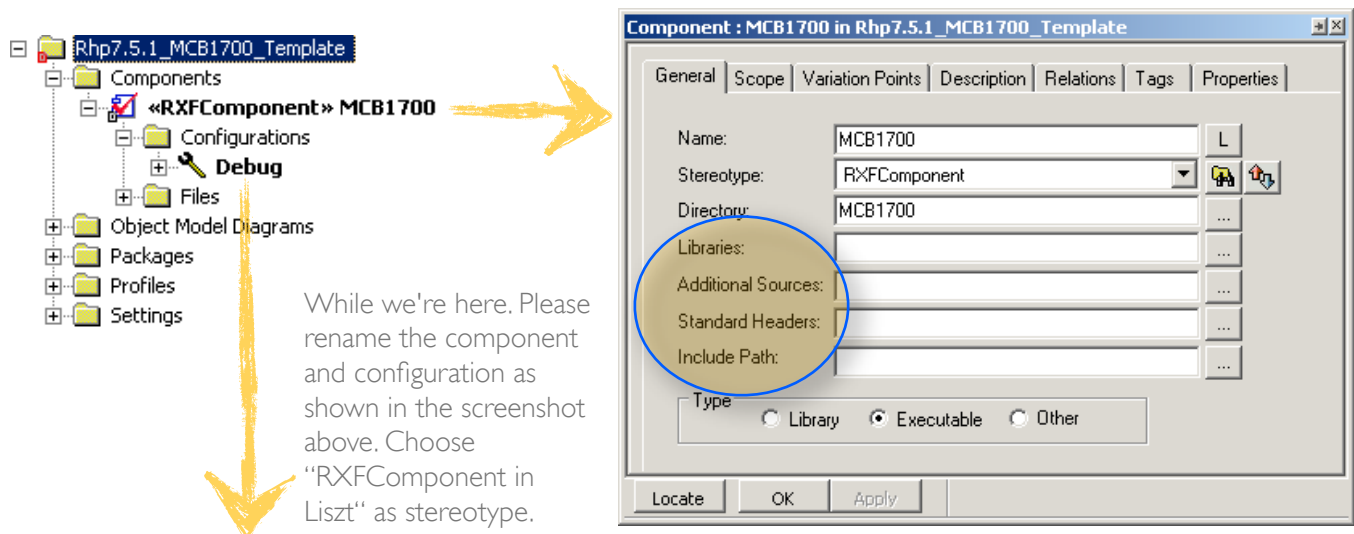With a bit more effort, of course, this is also possible.

Another major topic is reverse engineering.
Reverse engineering is often used, when old projects should be transferred into an actual UML model. Rhapsody will translate our sources automatically into UML elements. Usually, we need to edit the newly created elements manually.
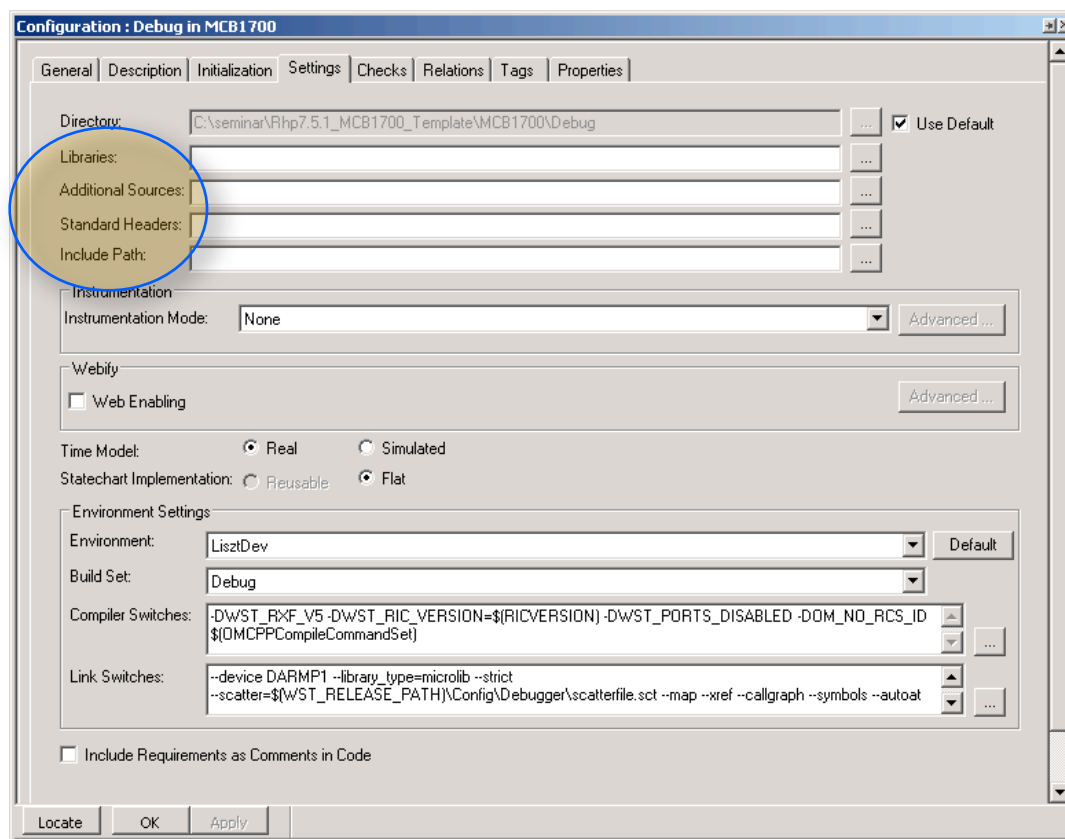So, it is a bit more semiautomatic ;-)

Component / Configuration

For Rhapsody beginners, there are two important points where we can implement all kinds of external code.

Component Features

While we're here. Please rename the component and configuration as shown in the screenshot above. Choose "RXFComponent in Liszt" as stereotype.

Configuration Features

Here is space, to include sources and libraries that are automatically included in the generated makefile. But why do we have a choice between the component and the configuration features? Often, libraries are build set dependent. These libs should be integrated into the configuration. Moreover, if you have multiple hardware platforms in a single project, you probably have separate components for each platform. Often, each platform needs different source files. Insert such files under component. You see, it makes sense.

Attention:
Separate multiple file names using commas, without spaces.

Properties

Sometimes, there is only one class in your project that needs an extra include. Open the Features with a double click and select the Properties-Tab. You will find an "ImpIncludes" entry under C_CG (C Code Generation). Insert your include here.

Every package has also the "ImpInclude" entry. So you can reach every class in a package with one entry. This is just an example, don't do it.
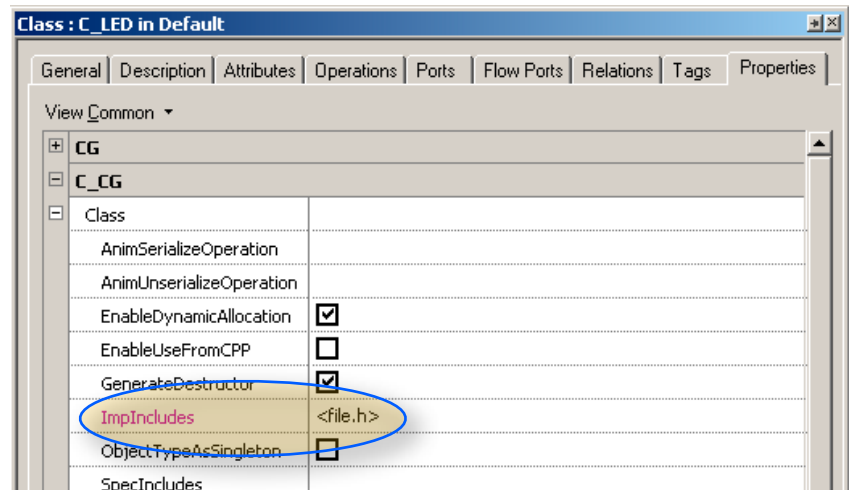

Files

The previous methods have the disadvantage, that the included files are not a part of the model. When a file is added under the component it is possible to generate it from Rhapsody and store it in the model.
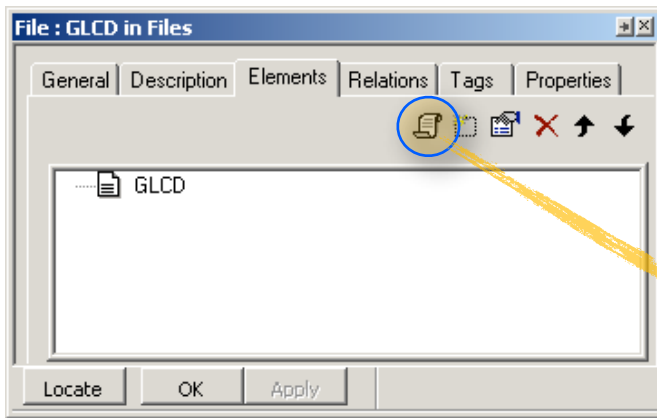
Make a right click on the component and select "Add New / File"

rename this file in "GLCD"

Info
Logical: *.c and *.h
Specification: *.h
Implementation: *.c
Other: no extension
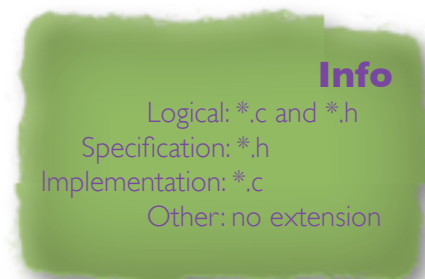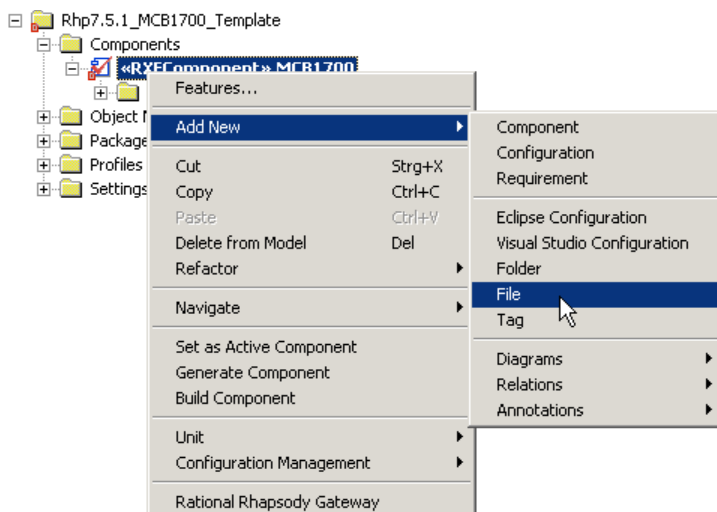
Open the features window from the GLCD file and select the general tab. Because we need a *.c extension, choose Implementation as type. After that, go to the Elements tab in the same window.
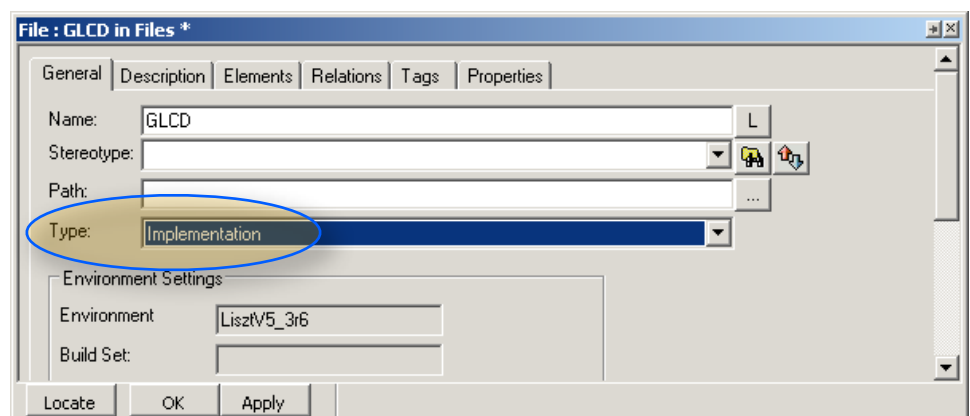
In the elements tab, you can add the content of the file that is stored in the model. Select the "New Text Element" symbol.

A new window is opening. We have prepared some content for the GLCD.c file. Look at

`Rhapsody\Share\WST_RXF_V5\Liszt\Source\MCB1700`

You will find a file called GLCD.c too. Open this file with a normal text editor. Copy the whole content to the clipboard and from the clipboard in this field.
Close both windows with OK.

The content of the GLCD.c file is now stored directly in your *.cmp file.

After the next code generate, you can browse this file by using active code view.
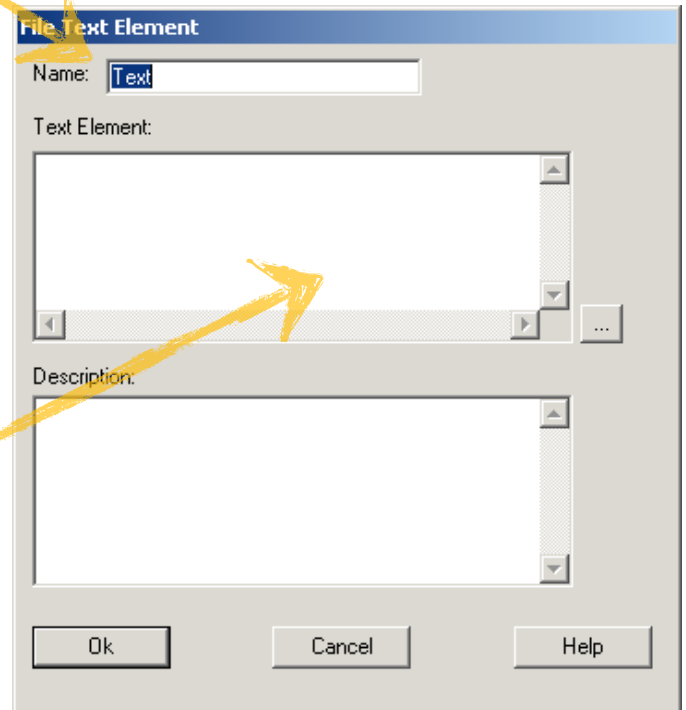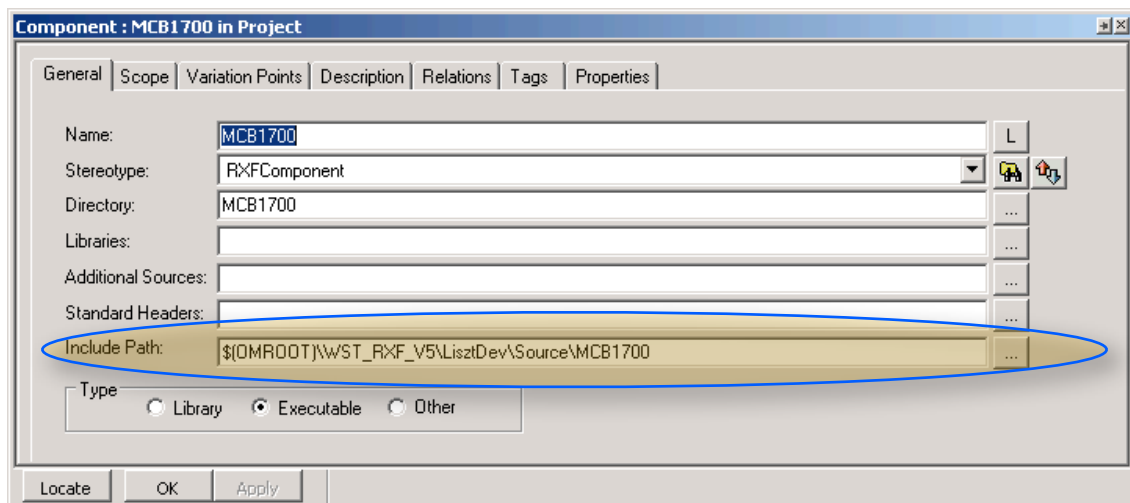
External Files

It is not required to add files to the project. We can also leave the files at their location. The next example shows us a cool method using external files. We will now add two external header files to our project. After that, we include these files by an object model diagram. Let's go...

First, Rhapsody must know, where our files are. Open the features dialog from the component. Insert the following path into the "Include Path" field:
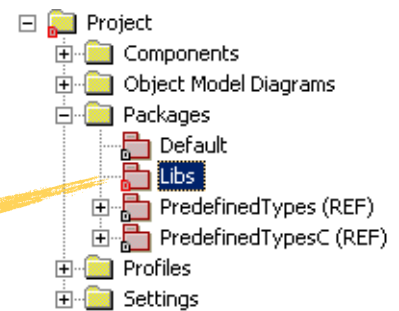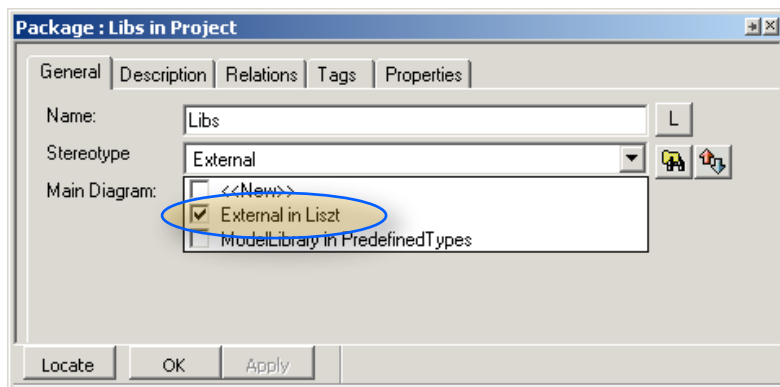
`$(OMROOT)\WST_RXF_V5\Liszt\Source\MCB1700`
`is equal to -> C:\Programme\IBM\Rational\Rhapsody\Share\WST_RXF_V5\Liszt\Source\MCB1700`

($(OMROOT) is a placeholder for share directory)

First, we need a new package for our files. Add a Package named "Libs" directly under Packages. Make a double click on this package and select "External in Liszt" as Stereotype.

Now, the package is marked as external. Open the Libs package and insert two more packages. One for the microcontroller header. The other for graphic display support.

Insert a file in each package by clicking "Add New / File". Rename the cortex file "lpc17xx" and the display file "glcd".
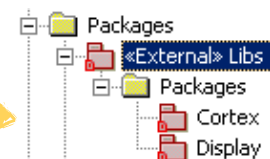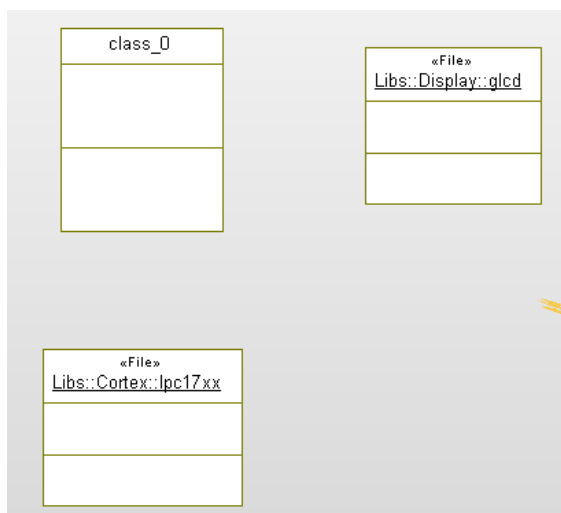
Select one of these files and take a look at active code view.

Include these files

We will now create a new class. Open the object model diagram. Select the "Class" symbol in the drawing toolbar and click into the object model diagram. Now, drag and drop the "lpc17xx" and the "glcd" files from the browser into the OMD. The result should look like this.

Draw dependencies from the class to each file. Open the features dialog from each dependency and select "Usage in PredefinedTypes" as stereotype.
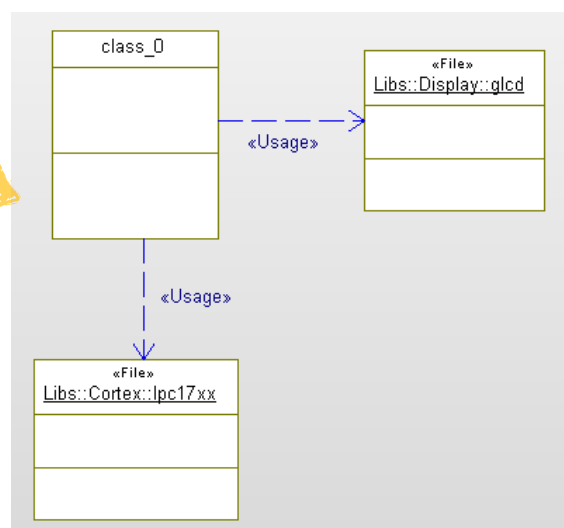

```
17  /*## dependency glcd */
18  #include <glcd.h>
19  /*## dependency lpc17xx */
20  #include <lpc17xx.h>
```

You will find two new includes in the class_0.h

Controlled Files

Controlled files are added to the model but are not automatically compiled. You can use it for manuals, pictures, or something that belongs to the model.

As an example, we will associate two pictures with our header files.
Make a right click on the "Cortex" package and select Add New / Annotations / Controlled File

That's what we want. Click OK.
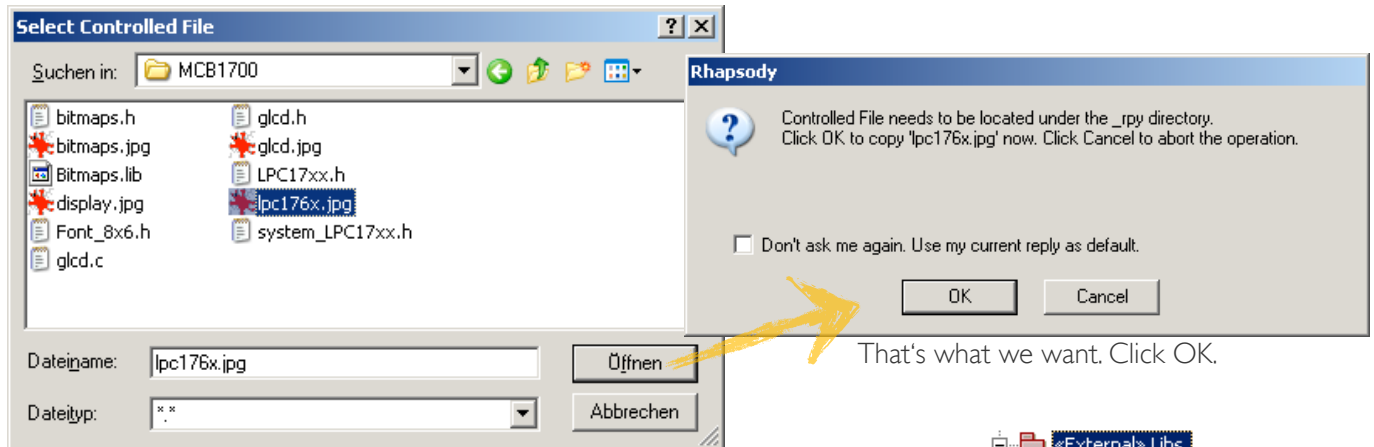
Do the same with the "Display" package and the "display.jpg" file.

Associate header files with these pictures

Go back to the OMD and make a right click on each header file. Select "Associate Image" from the contextual menu. You will find these two files directly in your Rhapsody project folder (Project\Project_rpy).
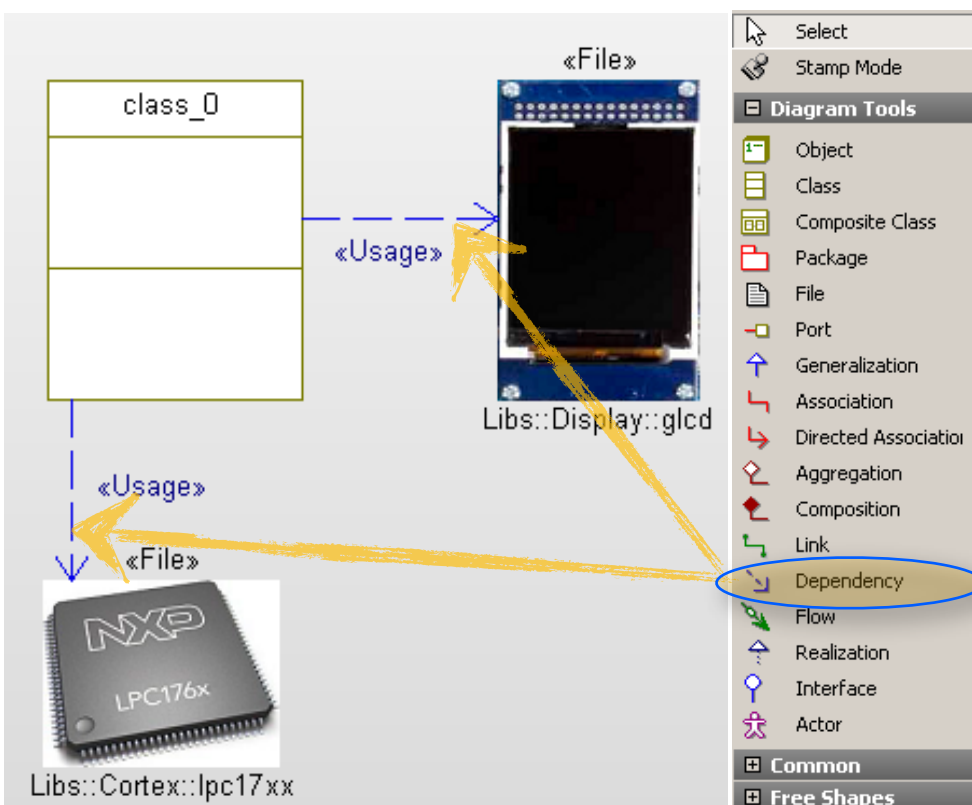

Info
You can remove the files from the diagram view. The association will still remain. Unfortunately, the dimensions will be reseted.

Reverse Engineering

Another possibility for integrating external sources is the use of external sources. There are 2 possibilities to use Reverse Engineering,

- You can use RE to integrate the scanned sources in the Rhapsody Model (Visualization and Code update)

- You can use RE to parse the sources and create links to them in your Rhapsody Model. (Visualization only)

Rhapsody in 'C' Reverse Engineering can parse source files (.c/.h) and even makefiles. It will create files that contain functions and variables. You can automatically create Object Model Diagrams. Other diagrams, like state-charts, cannot be drawn automatically.
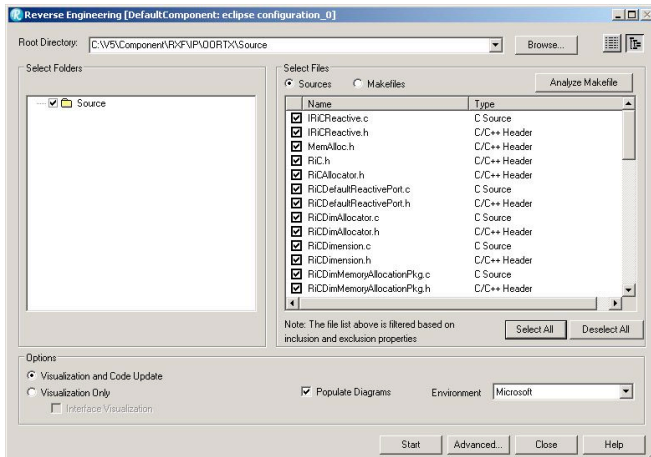

You will find RE under "Tools" - "Reverse Engineering"

Select hat and the window in the picture on the left will appear. here you can select the files that you want to include in reverse engineering.

You can select .mak, .c and .h files. If you selected the files to RE then you have to adjust the settings. This is done with the "Advanced" button.

In the advanced menu the 'C'-code parser is programmed so that it knows what to Reverse Engineer and what is must do with the result.
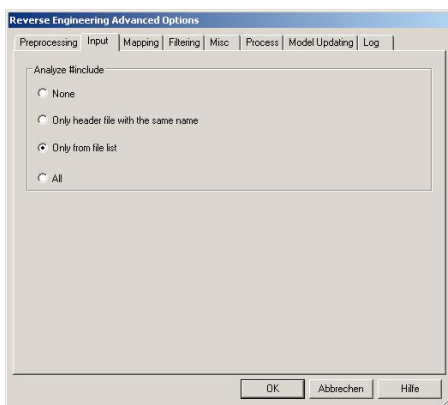

There are multiple tabs in the settings window, a quick overview

- Preprocessing. Use this to set or unset Defines (-D/-U on the Command Line) and to set Include Paths where to look for include files. (-I on the CL)

- Input. Here is set what to do with include files. Either all are parsed (beware, you include more then you think!), just the ones with the same name, just the ones explicitly named or none at all.

- Mapping. Extra field "Visualization only" that sets the mode where just links to external sources are included in the model. You can set where in the model tree you want to store the RE files.

- Reference Classes, only for C++/Java

- Model as Language Types, is the possibility to tailor the scanner to recognize non-ANSI keywords.

- Process. What to do when errors are found, abort, continue, report.

- Model Updating, just the option to automatically generate an Object Model Diagram

- Log, what is logged, what is scanned, what is updated.

When the "Start" Button is pressed, Rhapsody will scan all sources to include the result in it repository. You can use all elements in your model, you can add descriptions and you can improve your scanned files by changing the model and then generate 'C' files again.
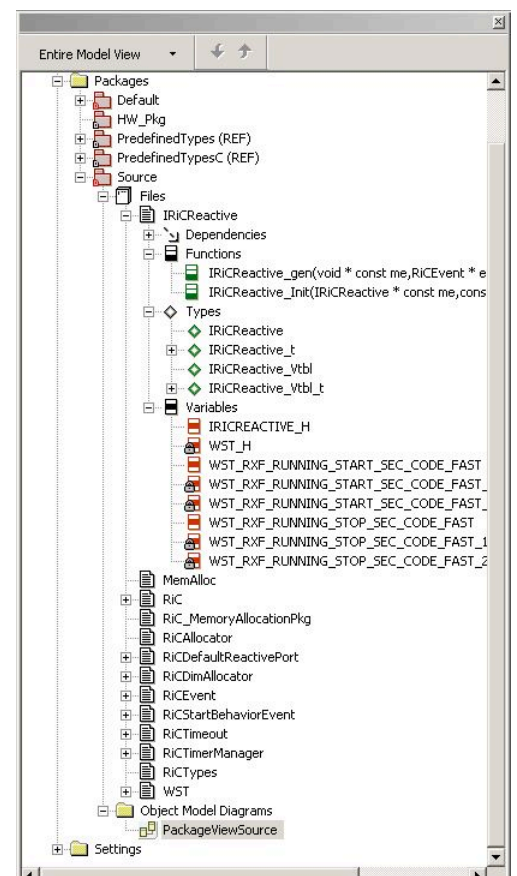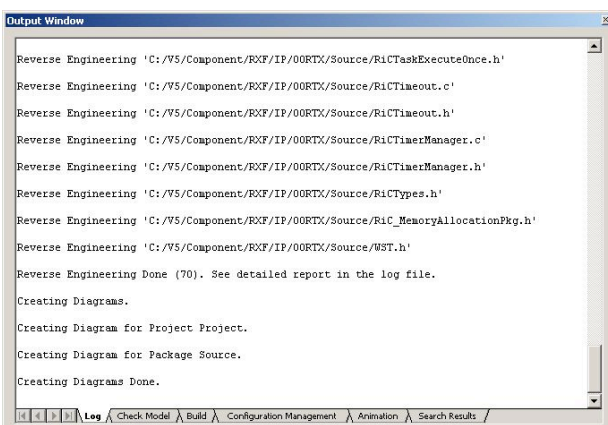
Product:

Embedded UML Start-Up Training

Author:

Marco Matuschek
Walter van der Heiden

Editor:

Willert Software Tools GmbH

Hannoversche Strasse 21

DE - 31675 Bückeburg

www.willert.de
info@willert.de

WILLERT.

IBM
Business
Partner