

net.betterdeveloper.understand Flavor for Structure 101g

By Marcio Marchini (marcio@BetterDeveloper.net)

2016/02/11

1) WHAT IS THE NET.BETTERDEVELOPER.UNDERSTAND FLAVOR FOR STRUCTURE101G?.....	1
1.1) LANGUAGES SUPPORTED	1
2) USING NET.BETTERDEVELOPER.UNDERSTAND	2
2.1) INSTALL UNDERSTAND	2
2.2) CREATE THE UNDERSTAND PROJECT	2
PROJECT NAME:	2
LANGUAGE:	2
ADD THE SOURCE DIRECTORIES:	3
CONFIGURE MORE SETTINGS	5
EXAMPLE OF EXTRA CONFIG:	5
LET UNDERSTAND ANALYZE	6
2.3) LOAD THE UDB INTO STRUCTURE 101G WITH THE PLUGIN	7
FLAVOR SELECTION.....	8
UDB SELECTION.....	8
MODEL LOADED INTO STRUCTURE 101G.....	9
3) RUNNING THE FLAVOR FROM THE CONSOLE.....	9
4) TIPS AND TROUBLESHOOTING	10
4.1) MERGING CPP AND H FILE PAIRS	10
4.2) FAILURE TO CONVERT / RUN THE FLAVOR.....	11
4.3) CHMOD +X.....	12

1) What is the net.betterdeveloper.understand Flavor for Structure101g?

This is a sort of a plugin for [Structure 101g](#) (called a flavor). It allows you to analyze projects in various programming languages by analyzing the Understand UDB files using Structure 101g.

1.1) Languages supported

Although Understand can analyze many different kinds of input, we only support the following ones:

- C/C++
- Pascal/Delphi
- Python

2) Using net.betterdeveloper.understand

You must install this flavor from the Structure101g Flavors menu¹.

2.1) Install Understand

In order to run this plugin, you will need a valid Understand license from <http://www.scitools.com/>. We only support the 64-bit versions.

NOTE: Make sure the version of Understand used to create the .UDB file is the exact same version/build-number as the version of the Understand udbapi.dll that you point at when converting.

2.2) Create the Understand Project

After launching Understand, use File->New->Project, and fill in the fields as shown below. We will use C++ as an example.

Project Name:



The image shows a dialog box for creating a new project. It has two main input fields. The first is labeled 'Name:' and contains the text 'FlightGear 2.4.0'. The second is labeled 'Directory:' and contains the text 'C:\temp'. To the right of the 'Directory:' field, there is a small downward-pointing arrow icon and a button with three dots '...', which typically indicates a file browser or dropdown menu.

Language:

Choose C/C++ in the case of the Flightgear project.

¹ Under Linux and MacOS you must make sure the executable has +x permissions (this is a limitation of the uncompression utility used by Structure101g when installing the flavor).

Ada
 COBOL (Beta)
 C/C++
 C#
 Fortran
 Java
 Jovial
 Pascal
 PL/M
 Python
 VHDL
 Web*

Use strict analyzer for C/C++/Objective-C/Objective-C++ (Beta)
 Import project settings from a MS Visual Studio Project (C/C++/C#)


Add The Source Directories:

Add a Project Directory [X]

Directory: Z:\nqm\temp-vmware\flightgear-2.4.0\flightgear-2.4.0\src [...]
 Configured Filters: C, C++ (*.c;*.C;*.H;*.cc;*.cpp;*.cxx;*.h;*.hh;*.hpp;*.hxx;*.inl) [...]
 Additional Filters: *.inc [...]
 Exclude: .* [...]

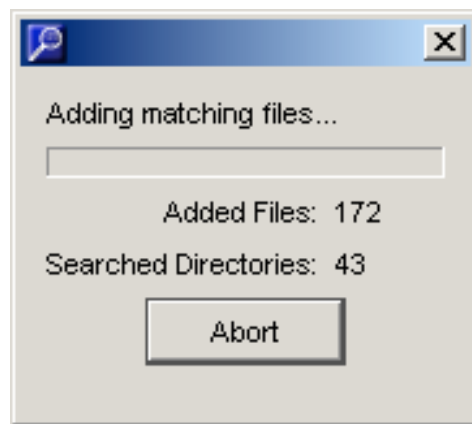
Include subdirectories

Watch this directory

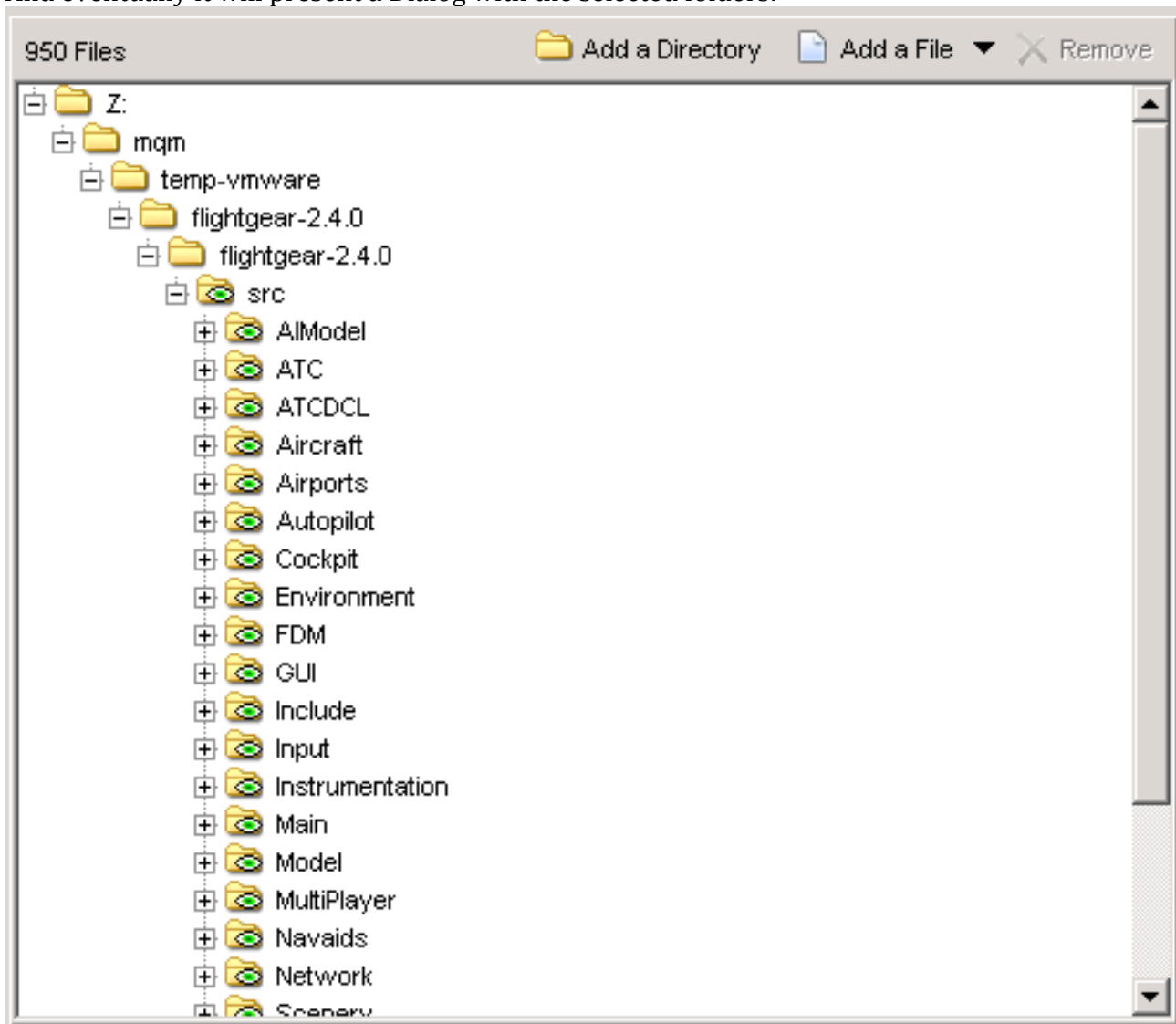

 The directory will be watched for any on disk changes. If a file is deleted from disk, it is removed from the project. If new sources are added to this directory or subdirectory of sources is added, it is automatically added to the project. Those files and directories must match above filters and settings.

OK Cancel

The tool will show its progress:



And eventually it will present a Dialog with the selected folders:



Configure More Settings

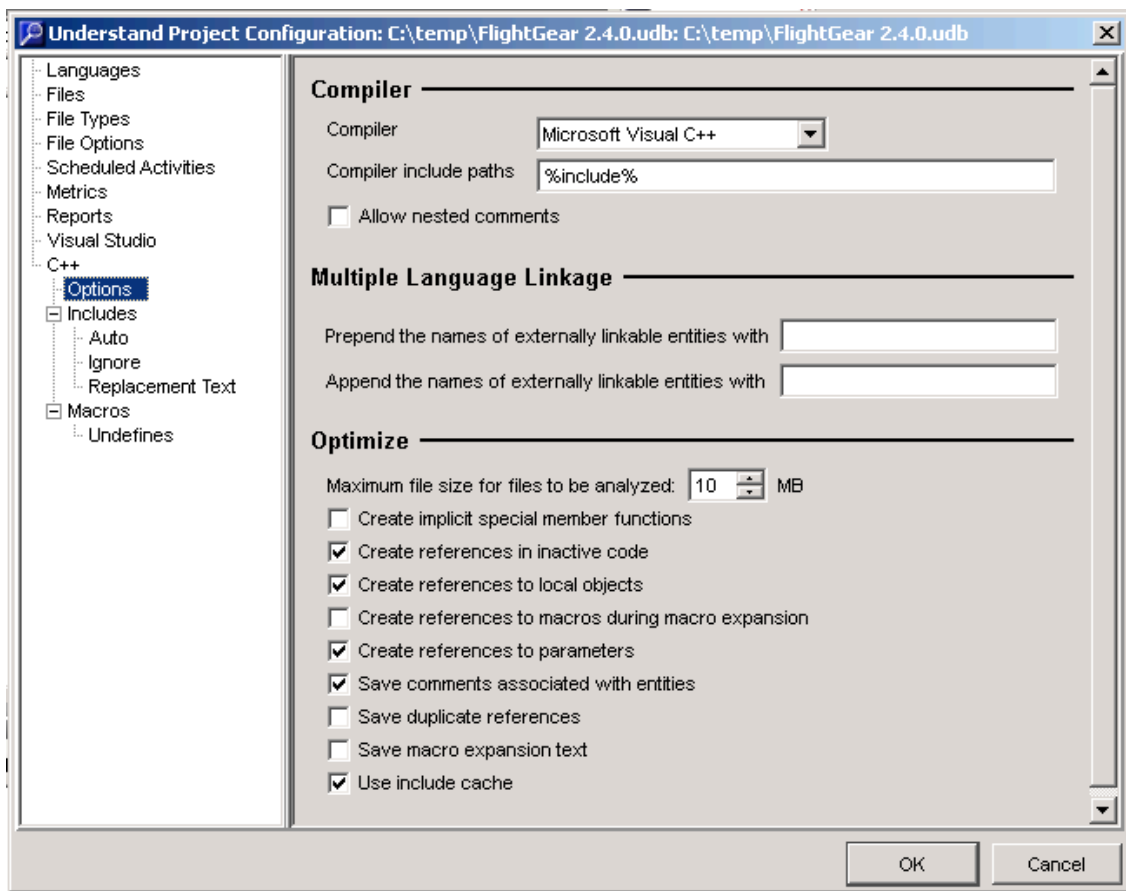
If you need to make extra tweaks to the project, make sure to select this non-default radio button at the end of the wizard:

This will allow us to configure some more options.

Example of Extra Config:

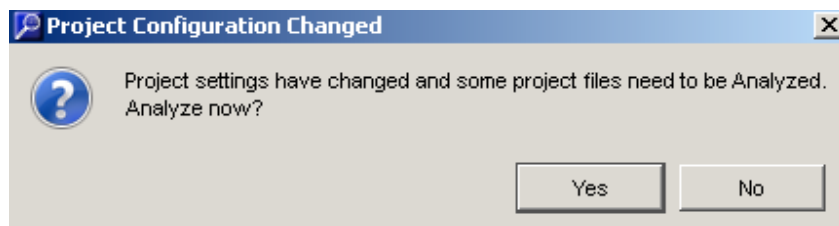


<input type="radio"/> Analyze project now
Begin analyzing project sources. During the analysis you may be prompted for optional configuration items. They may add further information to your parsed results but are not required and can be safely ignored.
<input checked="" type="radio"/> Configure more settings
Present the Project Configuration Dialog to view language specific configurations. All required configurations have been initialized to most common defaults. This can be invoked later at any time from the Project menu

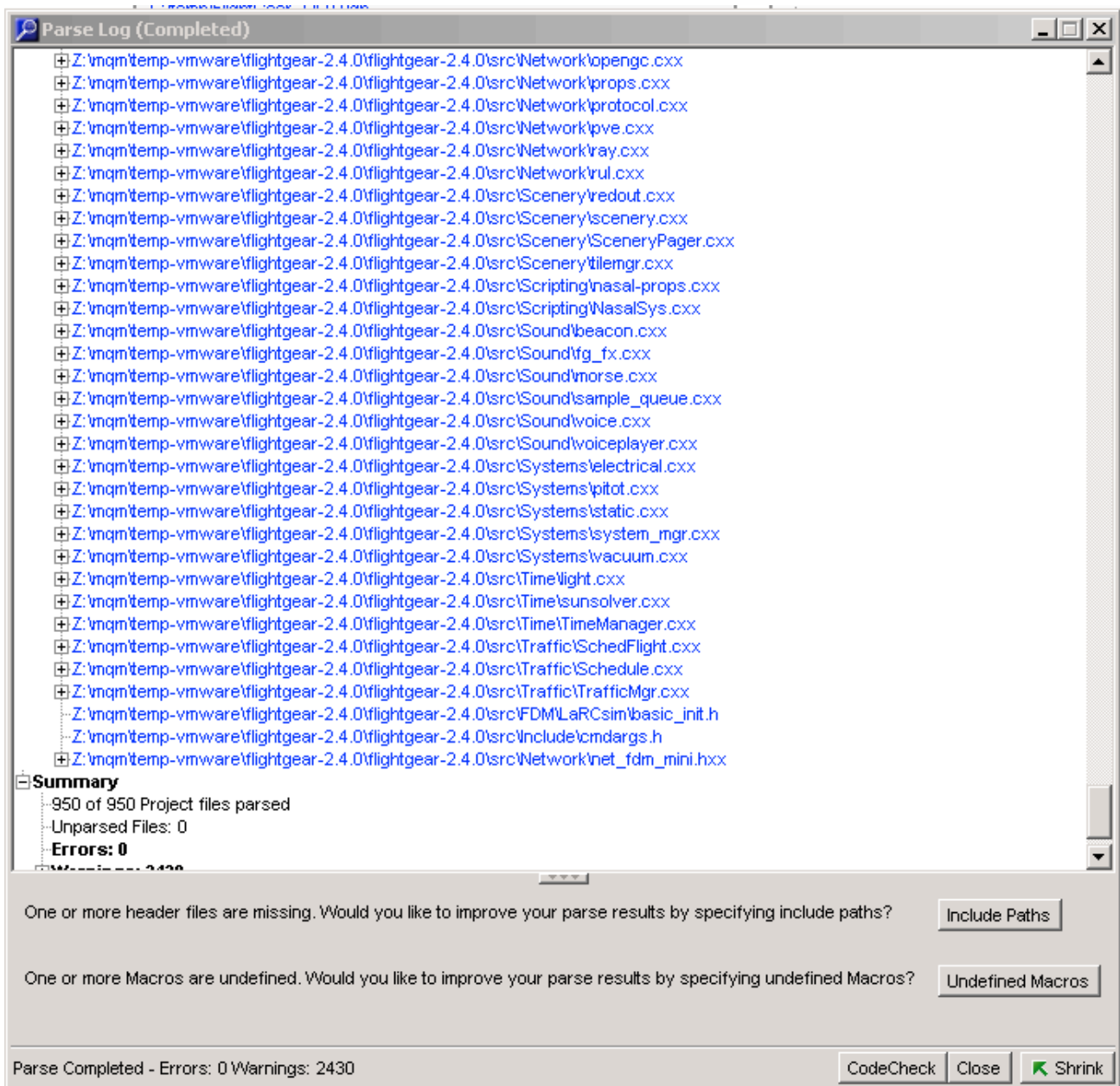


Let Understand Analyze

Now you can click OK and confirm when Understand asks to Analyze your files (if you changed the config options):



You should get a dialog showing progress, until it finishes:



2.3) Load the UDB into Structure 101g with the Plugin

Once you have a UDB file, you can load it into Structure101g on any computer with a valid license². Load Structure 101g and choose File->New, then select our plugin (flavor):

² You will need a license for SciTools Understand, a license for Structure101g and a license to run this flavor.

Flavor Selection

Project type



Project type:

Flavor: **net.betterdeveloper.understand**
Version: **1.0.1**
Dependencies as reported by Understand's UDB file
Parameters: *UDB File, Files to keep (CSV of regexes), Directory where Understand binary is, ...*

Click Next and make sure to point at the UDB file you created (/Users/mqm/Downloads/flightgear.udb in our case):

UDB Selection

Project Parameters

Specify the parameters required to load your **net.betterdeveloper.understand** project below.

Simply (double) click the appropriate parameter value to set it. To get more information about the meaning of a particular parameter, mouse over the relevant cell to get a tooltip.

Parameter name	Parameter value
UDB File	/Users/mqm/Downloads/flightgear.udb
Files to keep (CSV of regexes)	.*
Directory where Understand binary is	/Applications/scitools/bin/macosx
Detailed traversal to the function/variable level	Yes
Mixed namespace/non-namespace	Yes

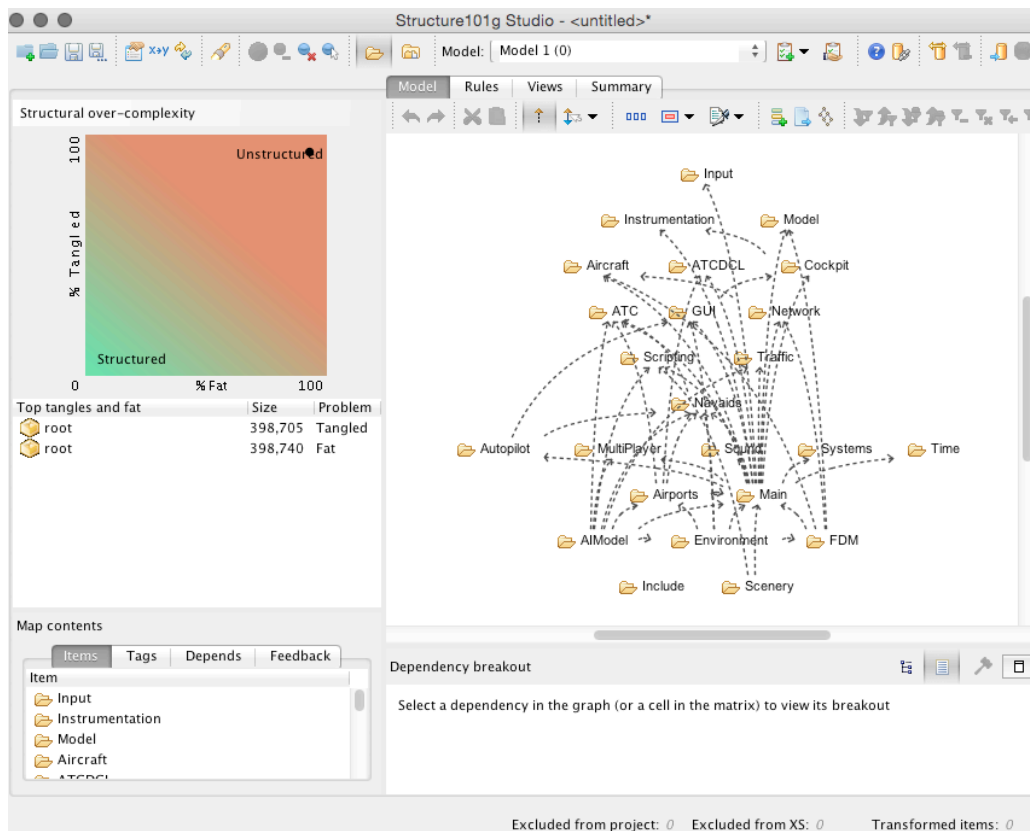
Note that the GUI allows you to tweak some other values:

- Files to keep: leave the default value `.*` until you become an advanced user. This is a regex that filters which files you really want to capture from the UDB file. It may be useful to tweak this parameter to filter out library files. For example, if all interesting files are under a folder called `MyProject`, you can use `MyProject` as the regex.

- Directory where Understand is: Point it at the directory where the und.exe executable is. By explicitly pointing at the Understand install dir you can support multiple versions of Understand without having to tweak the global PATH environment variable.
- Detailed traversal to the function/variable level: Leaving this off will make the converter run faster and capture dependencies at the file level. Turning it on will produce a much bigger model, discriminating the various dependencies at the function and variable types level.
- Mixed namespace/non-namespace: This is most useful with C++ input. Some languages use namespaces as **the** mechanism to organize modules. Others use directories in the file system. Most C++ sources use a mix of namespaces and directories. Therefore, when traversing the sources, the flavor will take into account this mixed mode. For other languages, leave this off.

After it processes the UDB file, you should get your model loaded, like below:

Model Loaded Into Structure 101g



You can now analyze your system using Structure 101g's great features.

3) Running the flavor from the console

This flavor comes with 3 subdirectories, one for each platform supported: MacOS, Linux and Windows. If you want to run the conversion from a command-line or a script, you can. In these cases

we highly recommend that you use the GUI wizard and use the little “Copy command line” link at the bottom/right of the Dialog. It will copy to the clipboard the full command-line used by the wizard behind the scenes. For instance, here's an example of what we used to run it on the FlightGear 2.4.0 C++ source files:

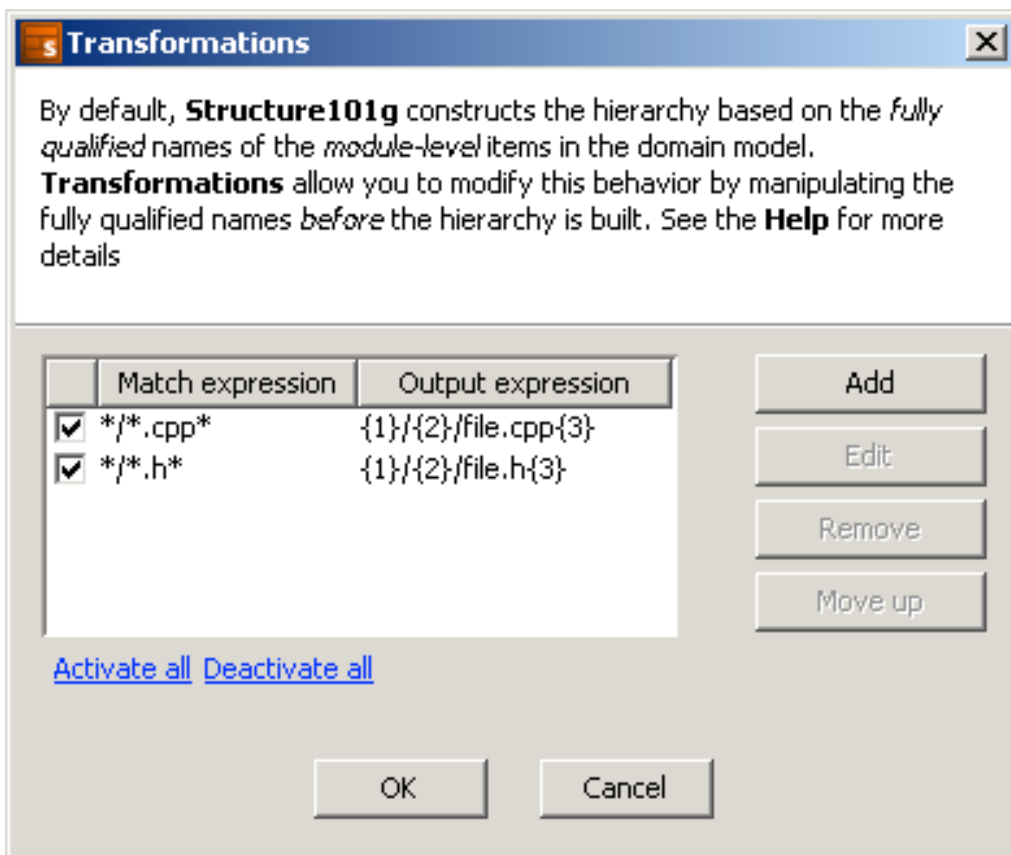
```
/Users/mqm/structure101g/flavors/net.betterdeveloper.understand_1.0.1/macos/und2s101 -i  
/Users/mqm/Downloads/flightgear.udb -o  
/var/folders/ts/c1cr6kcd4yb6s7tj6bjt_nyr0000gp/T/s101g_6529962784247109001.tmp -k.* -u  
/Applications/scitools/bin/macosx -t  
/Users/mqm/structure101g/flavors/net.betterdeveloper.understand_1.0.1 -d true -m true
```

Try this approach different ways until you become familiar with the command-line options. Then you can invoke the executable on your own customizing the parameters as you want.

4) Tips and Troubleshooting

4.1) Merging CPP and H file pairs

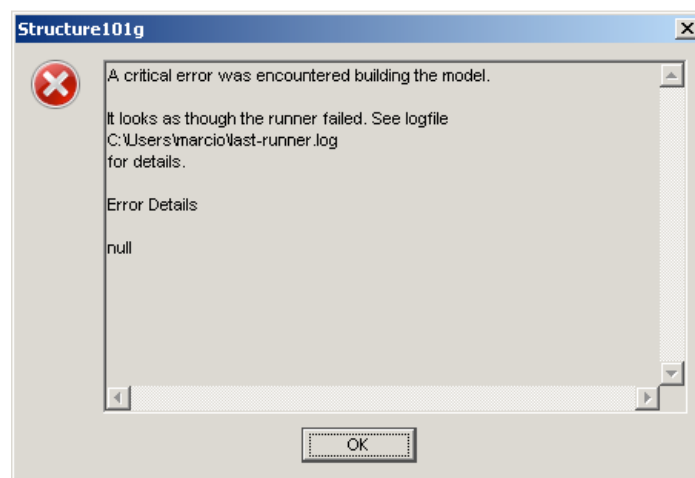
Most of the time, H files and CPP files are used in pairs, and the dependencies can become a bit awkward because you have two separate entities in the Structure 101 GUI. In these cases, you may want to use Structure 101's Model->Transformations, and group pairs of H and CPP files together as a single entity. The example below creates a virtual folder with both files:



Note that files such as foo.h and foo.cpp will be put together under a (new logic) folder called foo: foo/foo.h and foo/foo.cpp. This comes in handy when analyzing some C++ systems.

4.2) Failure to Convert / Run the Flavor

In some cases the flavor may fail to run. In these cases you will get an error dialog like this:



If you open the log file shown, you should find the reason. Example:

[Err] Error during conversion: 3

The possible errors and their causes are:

- **3:** The flavor could not open the UDB file. Please make sure your local copy of Understand, in the PATH, can load this file without problems (details below). In most cases this error happens when a UDB file created with an older Understand version is being loaded, without upgrading the file first using Understand itself (our flavor cannot do this automatic conversion). Another case is when you don't have enough floating licenses for Understand (running the conversion consumes 1 Understand license). In order to make sure you can run this flavor, do this:
 - Open a command prompt ("DOS")
 - Execute: *und -db yourDbFile.db*
 - Make sure it can load the file without errors and without asking to re-analyze sources. If it needs to reanalyze/rebuild/reparsed, our flavor will fail.
 - The complete set of possible errors is:
 - DBAlreadyOpen - only one database may be open at once
 - DBCorrupt - bad database file
 - DBOldVersion - database needs to be rebuilt
 - DBUnknownVersion - database needs to be rebuilt
 - DBUnableOpen - database is unreadable or does not exist
 - NoApiLicense - Understand license required
- **1:** The flavor could not find the Understand DLLs. Are you sure Understand is in the PATH for the user running the process that is performing the conversion?

4.3) chmod +x

Under Linux and MacOS you must make sure the executable has +x permissions (this is a limitation of the uncompression utility used by Structure101g when installing the flavor).