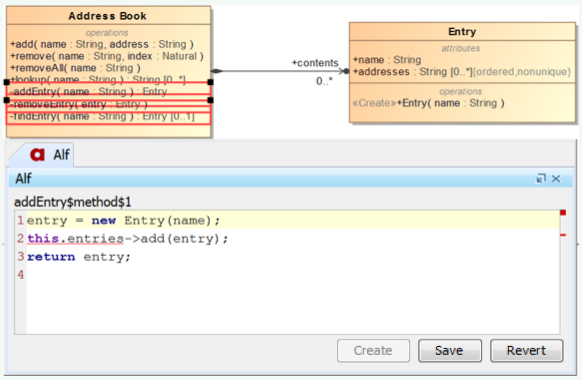
Dependency Management

Dependency Tracking

Alf code usually references various elements of the model that contains the Alf text. The Alf compiler keeps track of such dependencies, so that it can tell you when a change in the model causes an error in dependent Alf code. When you first open a project with Alf code in it, the Alf compiler will automatically parse all the code in the project, in order to re-establish the dependencies of the code on other model elements.

You can cancel the processing of Alf code on project open, but, if you do, dependencies will only be established for code that has been fully parsed up to that point. The Alf compiler will then be unable to determine when it is necessary to recompile code for which dependencies have not been established. Therefore, do not cancel the dependency processing on project open if you intend to make any changes to the model you are opening. If you do cancel the initial processing, you can later re-establish all dependencies by using the **Tools > Alf > Clean Project** command to do a clean rebuild of the project.

For example the figure below shows a class diagram from the Address Book sample model, in which the Association end name *entries* has been changed to *contents.* The Alf compiler has detected that the Alf code for the Operations *addEntry, removeEntry and findEntry* depend on this Association end, and, so, the compiler has marked that these Operations now have compilation errors. The Alf code for *addEntry* is shown in the Alf editor, showing the error on the reference *this.entries.* Changing this reference to *this.contents* eliminates the error for this Operation.

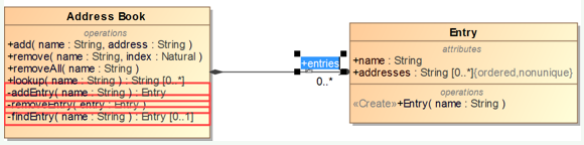


Dependency tracking example

Rebuilding manually

Sometimes, the Alf compiler will lose track of dependencies and will not be recognize that an element marked as having Alf compilation errors could be recompiled successfully. This may happen, for example, if you delete a referenced element and create a new element with the same name, or if you use **Undo** to reverse the change that cause the errors in the first place. In such cases, you can manually initiate recompilation using the **Tools > Alf > Build Project** command to rebuild your project.

If, in the previous example, suppose, instead of changing the Association end name, that you delete the Association. Then you create a new Association, with the same end name *entries.* Even though the end name is the same, it is a different model element than in the old Association, so the Alf compiler has no record of any code dependency on it. However, if you do an Alf build of the project, then the code compiles successfully.



Example with a new Association with the same end name, but still needing to be rebuilt

Table of Contents

- Dependency Tracking
- Rebuilding manually
- Turning off automatic building

Related pages

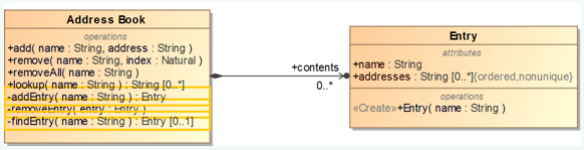
- The Alf compiler
- Building and cleaning
- Environment options

If you change some Alf code with compilation errors to correct those errors, and then **Undo** that change, the compilation error annotation may not be restored. In this case, the Alf compiler will not recognize the need to recompile the code on a normal Alf build of the project, so the code will not be marked as having compilation errors, even though it is erroneous. In order to restore the proper error annotation, use the **Tools > Alf > Clean Project** command to do a clean build of the entire project.

Turning off automatic building

If you don't want the Alf compiler to automatically rebuilding your project when it detects code that needs to be compiled, you can set the Alf environment option **Build Automatically** to **false**. If you do this, the Alf compiler will mark elements that need to be recompiled with a warning annotation, rather than automatically recompiling them. This may be useful if you want to make several model changes outside the Alf code, and then recompile all the Alf code at once. To force actual recompilation of the Alf code for elements marked as needing recompilation, use the **Tools > Alf > Build Project** command to rebuild the project.

If the association end in the Address Book example is changed with the **Build Automatically** option turned off, then the affected operations are marked with warning annotations as shown below. Rebuilding the project then results in error annotations, as shown previously.



Example showing recompilation-needed warning annotations