

# Importer API: Parsing

An Alf model unit file may be initially imported using the *AlfImporter.parse* method, passing in a Java *Path* object for the file. The file must be in the model directory identified when the *AlfImporter* object was [created](#). The Alf code in the given file is then parsed and the resulting abstract syntax tree is cached. Any subunits of the Alf unit in the given file are also imported. In addition, when the import process identifies a reference that cannot be resolved within the file being imported (or within the UML model being imported into), it will attempt to find another associated file in which the reference can be resolved. The importer uses the conventions of the Alf Reference Implementation when resolving Alf unit names to file names, starting from the model directory identified for the importation (for more information, see the Alf Reference Implementation documentation [here](#)).

The *parse* method returns a Boolean indicating whether all parsing completed successfully or not. You can also check the success of the last parse by calling the *isSuccessful* operation. You can call the *parse* method multiple times to load multiple different files (which must all be from the same model directory) into the import cache. Note that the *isSuccessful* flag is reset after each parse operation.

Unlike parsing using the *AlfCompiler*, the parse process in the *AlfImporter* does *not* include constraint checking. Instead, constraint checking is done as part of the [mapping](#) process, on all cached model units and their subunits together. This means that any errors reported from parsing imported code will be syntactic errors. All such errors are reported as messages in the MagicDraw [Message Window](#). It is also possible for the parse to be unsuccessful without reporting any syntax errors, for one of the following reasons:

- The identified file is not found or attempting to read it results in an IO exception (though no exception is propagated).
- The identified file has a valid Alf unit in it, but the name of that unit does not match the name of the file.

```
AlfImporter importer = new AlfImporter(modelDirectory, progressStatus);
if (importer.parse(Paths.get(modelDirectory, "Utilities.alf"))) {
    importer.parse(Paths.get(modelDirectory, "Main.alf"));
}
```