Importer API

The *AlfImporter* class provides the basic interface for importing Alf code from external files. An *AlfImporter* is instantiated with two pieces of information:

1. A string giving the path to the *model directory,* which is the root directory for all files being imported.
2. An optional *ProgressStatus* object, the description of which is updated with the names of files being imported. (This argument may be null, if you do not wish to display progress status.)

Importing of Alf code then takes place in two steps:

1. Parsing one or more Alf model units (along with their subunits) from files in the model directory, caching their resulting abstract syntax representations.
2. Mapping (after constraint checking) all parsed units into the UML model in MagicDraw.

The API allows you to carry out these steps individually (which is useful if, for example, you wish to import multiple model unit files before mapping) or using a single call (for a single model unit file).

> ⚠️ Importation takes place into the currently open and active Project. Once you instantiate an *AlfImporter* object in a Project, it is no longer consistent to use an *AlfCompiler* object in that Project. Instead, once you have completed any importation, you must use the *AlfActionUtil,. resetActiveProject* method to return the currently active project to a state in which the *AlfCompiler* can be used (see Importer API: Mapping). Alternatively, you can use the *AlfActionUtil. importFrom* method to do the importation (as described in Importer API: Utilities), in which case the Project will be restored for normal Alf operations at the completion of the method.