

# Importing Alf

Alf may be used in MagicDraw to textually specify detailed behavior. However, the Alf specification also provides an *extended* level that includes notation for the static modeling constructs of Foundational UML, including packages, classes, data types, associations and signals. An Alf representation such as this may be imported from a file into MagicDraw, creating the specified model elements and inserting Alf bodies as appropriate.

To import an Alf file from the Welcome Screen:

1. From the Welcome Screen (i.e, with no projects open), select **File > Import From > Alf File**.
2. Choose the file to be imported and click **Open**.
3. A new, untitled project is created, and the selected Alf file is imported into it.
4. Save the project.



If there are any syntactic errors or constraint violations in the Alf being imported, the import will fail and no model will be created in MagicDraw. The Notification Window will open to show the compilation errors.



It is possible for a model to be represented by multiple Alf files that reference each other. In this case, you should select the root file as the one to import. When the import process identifies a reference that cannot be resolved within the file being imported, it will attempt to find another associated file in which the reference can be resolved. When importing, the Alf plugin uses the conventions of the Alf Reference Implementation when resolving Alf unit names to file name, considering the directory of the selected file to be the "model directory" (for more information, see the Alf Reference Implementation documentation [here](#)).

To import an Alf file into an existing project:

1. Open the existing project you want to import into.
2. Select **File > Import From > Alf File**.
3. Choose the file to be imported and click **Open**.
4. The selected Alf file will be imported, with new model elements created within the open project.



If the Alf in the imported file contains external references, then the Alf compiler will attempt to resolve these against existing elements within the model, before searching for another file from which to resolve them.



If you have projects open, and you want to import Alf into a new project, but don't want to close the currently open projects, then first use **File > New Project** to create a [new Alf project](#), and import the Alf file into that, as above.

As an example, consider the following simple *Address Book Model* represented in Alf:

```

package 'Address Book Model' {

    public class Entry {
        public name: String;
        public address: String;

        @Create public Entry(in name: String, in address: String) {
            this.name = name;
            this.address = address;
        }
    }

    public assoc AddressBook_Entry {
        public addressBook: AddressBook;
        public entry: compose Entry[*];
    }

    public class AddressBook {
        public add(in name: String, in address: String) {
            entry = this.entry->select e (e.name == name)[1];
            if (entry == null) {
                this.entry->add(new Entry(name,address));
            } else {
                entry.address = address;
            }
        }

        public get(in name: String): String[0..1] {
            return this.entry->select e (e.name == name)[1].address;
        }
    }
}

```

Importing the example above into MagicDraw results in the model structure shown below. Note that not diagrams are created by the import process. You will need to create any diagrams by hand once the import is completed.

Containment

Model

- Address Book Model
  - Relations
    - Association:AddressBook\_Entry[entry:Entry - addressBook:AddressBook]
  - AddressBook
    - add\$method\$1( name : String [1], address : String [1] )
    - AddressBook\$initialization\$1
      - AddressBook\$method\$1() : AddressBook [1]
      - destroy\$method\$1
      - get\$method\$1( name : String [1] ) : String [0..1]
      - #AddressBook\$initializationFlag\$1 : Boolean [0..1]
      - +AddressBook\$initialization\$1()
      - +add( name : String [1], address : String [1] )
      - +get( name : String [1] ) : String [0..1]
      - +AddressBook() : AddressBook [1] «Create»
      - +destroy() «Destroy»
  - Entry
    - destroy\$method\$1
    - Entry\$initialization\$1
      - Entry\$method\$1( name : String [1], address : String [1] ) : Entry [1]
      - #Entry\$initializationFlag\$1 : Boolean [0..1]
      - +name : String [1]
      - +address : String [1]
      - +Entry\$initialization\$1()
      - +Entry( name : String [1], address : String [1] ) : Entry [1] «Create»
      - +destroy() «Destroy»

Alf

```
add$method$1
1  entry = this.entry->select e (e.name == name) [1];
2  if (entry == null) {
3    this.entry->add(new Entry(name,address));
4  } else {
5    entry.address = address;
6  }
7
```

Save Revert

Imported Address Book Model