

# Compiler API: Utilities

Instead of explicitly creating an *AlfCompiler* object and using its methods, you can alternatively parse and map Alf code using static methods provided by the *AlfActionUtil* class.

The *AlfActionUtil.parse* method takes a Named Element and, using this as the context element, parses and constraint checks the Alf body of the Element, if it has one. If the parsing and constraint checking succeed, then the method also automatically registers dependencies. The method returns a Boolean indicating whether the parse succeeded or not. If the parse fails, Compiler Errors are reported in an error message that is attached to the context element using an error Annotation (which then becomes available as an Active Validation Result, as described in [the Alf editor](#)).

```
if (AlfActionUtil.parse(element)) {  
    // Parse was successful.  
    ...  
}
```

The *AlfActionUtil.compile* method takes a Named Element and, using this as the context element, both parses and maps the Alf body of the Element, if it has one. If the compilation is successful, then the method updates the context element and automatically registers dependencies. The method returns a Boolean indicating whether the parse succeeded or not. As with the *parse* method, if the compilation fails, Compiler Errors are recorded in an error message that is attached to the context element using an error Annotation. Unlike the *AlfCompiler.map* and *AlfCompiler.compile* methods, you do not need to explicitly start a MagicDraw session to call the *AlfActionUtil.compile* method, because it ensures internally that a session is started for mapping, if necessary.

```
if (AlfActionUtil.compile(element)) {  
    // Compilation was successful.  
    ...  
}
```

## Related Pages

- [Compiler API: Parsing](#)
- [Compiler API: Mapping](#)