

Import

```
import TARGET (from ASSEMBLY)? (as ALIAS)?
```

The *import* construct makes all the members of the imported target available to the current module. So instead of writing:

```
System.Console.WriteLine("and now for something completely different... ")
System.Console.WriteLine("import!")
```

one can write:

```
import System

Console.WriteLine("anfscd...")
Console.WriteLine("import!")
```

The target can be either a namespace or a type. When it's a type, all the type's static members can be referenced directly by name. So the previous example could be simplified even further:

```
import System.Console

WriteLine("anfscd...")
WriteLine("import!")
```

The *from* clause can be used to specify an additional assembly reference as well as to disambiguate namespaces. When using a Namespace that is not defined in an assembly with the name of the Namespace, you **should** use *from*:

```
import Some.Namespace from Weird.Assembly.Name
import Gtk from "gtk-sharp"
```

from also accepts a quoted string as argument for weird named assemblies:

```
import Gtk from "gtk-sharp"

Application.Init()
```

And speaking of assembly references, the boo compiler automatically add 4 assembly references before compiling any code: Boo, Boo.Lang.Compiler, (ms)corlib and System.