

Chapter 41: Swift Package Manager

Section 41.1: Creation and usage of a simple Swift package

To create a Swift Package, open a Terminal then create an empty folder:

```
mkdir AwesomeProject cd AwesomeProject
```

And init a Git repository:

```
git init
```

Then create the package itself. One could create the package structure manually but there's a simple way using the CLI command.

If you want to make an executable:

```
swift package init --type executable
```

Several files will be generated. Among them, *main.swift* will be the entry point for your application.

If you want to make a library:

```
swift package init --type library
```

The generated *AwesomeProject.swift* file will be used as the main file for this library.

In both cases you can add other Swift files in the *Sources* folder (usual rules for access control apply).

The *Package.swift* file itself will be automatically populated with this content:

```
import PackageDescription

let package = Package(
    name: "AwesomeProject"
)
```

Versioning the package is done with Git tags:

```
git tag '1.0.0'
```

Once pushed to a remote or local Git repository, your package will be available to other projects.

Your package is now ready to be compiled:

```
swift build
```

The compiled project will be available in the *.build/debug* folder.

Your own package can also resolve dependencies to other packages. For example, if you want to include "SomeOtherPackage" in your own project, change your *Package.swift* file to include the dependency:

```
import PackageDescription

let package = Package(
    name: "AwesomeProject",
    targets: [],
```

```
dependencies: [  
    .Package(url: "https://github.com/someUser/SomeOtherPackage.git",  
             majorVersion: 1),  
]  
)
```

Then build your project again: the Swift Package Manager will automatically resolve, download and build the dependencies.
