

# The Move Book

The Move.toml is a manifest file that describes the [package](#) and its dependencies. It is written in [TOML](#) format and contains multiple sections, the most important of which are [package] , [dependencies] and [addresses] .

The [package] section is used to describe the package. None of the fields in this section are published on chain, but they are used in tooling and release management; they also specify the Move edition for the compiler.

The [dependencies] section is used to specify the dependencies of the project. Each dependency is specified as a key-value pair, where the key is the name of the dependency, and the value is the dependency specification. The dependency specification can be a git repository URL or a path to the local directory.

Packages also import addresses from other packages. For example, the Sui dependency adds the std and sui addresses to the project. These addresses can be used in the code as aliases for the addresses.

Starting with version 1.45 of the Sui CLI, the Sui system packages ( std , sui , system , bridge , and deepbook ) are automatically added as dependencies if none of them are explicitly listed.

Sometimes dependencies have conflicting versions of the same package. For example, if you have two dependencies that use different versions of the Example package, you can override the dependency in the [dependencies] section. To do so, add the override field to the dependency. The version of the dependency specified in the [dependencies] section will be used instead of the one specified in the dependency itself.

It is possible to add [dev-dependencies] section to the manifest. It is used to override dependencies in the dev and test modes. For example, if you want to use a different version of the Sui package in the dev mode, you can add a custom dependency specification to the [dev-dependencies] section.

The [addresses] section is used to add aliases for the addresses. Any address can be specified in this section, and then used in the code as an alias. For example, if you add alice = "0xA11CE" to this section, you can use alice as 0xA11CE in the code.

The [dev-addresses] section is the same as [addresses] , but only works for the test and dev modes. Important to note that it is impossible to introduce new aliases in this section, only override the existing ones. So in the example above, if you add alice = "0xB0B" to this section, the alice address will be 0xB0B in the test and dev modes, and 0xA11CE in the regular build.

The TOML format supports two styles for tables: inline and multiline. The examples above are using the inline style, but it is also possible to use the multiline style. You wouldn't want to use it for the [package] section, but it can be useful for the dependencies.

## Sections

The [package] section is used to describe the package. None of the fields in this section are published on chain, but they are used in tooling and release management; they also specify the Move edition for the compiler.

The [dependencies] section is used to specify the dependencies of the project. Each dependency is specified as a key-value pair, where the key is the name of the dependency, and the value is the dependency specification. The dependency specification can be a git repository URL or a path to the local directory.

```
```bash
```

## git repository

```
Example = { git = "https://github.com/example/example.git", subdir = "path/to/package", rev = "framework/testnet" }
```

## local directory

```
MyPackage = { local = "../my-package" } ```
```

Packages also import addresses from other packages. For example, the Sui dependency adds the std and sui addresses to the project. These addresses can be used in the code as aliases for the addresses.

Starting with version 1.45 of the Sui CLI, the Sui system packages ( std , sui , system , bridge , and deepbook ) are automatically

added as dependencies if none of them are explicitly listed.

Sometimes dependencies have conflicting versions of the same package. For example, if you have two dependencies that use different versions of the Example package, you can override the dependency in the [dependencies] section. To do so, add the override field to the dependency. The version of the dependency specified in the [dependencies] section will be used instead of the one specified in the dependency itself.

```
bash [dependencies] Example = { override = true, git = "https://github.com/example/example.git",  
subdir = "crates/sui-framework/packages/sui-framework", rev = "framework/testnet" }
```

It is possible to add [dev-dependencies] section to the manifest. It is used to override dependencies in the dev and test modes. For example, if you want to use a different version of the Sui package in the dev mode, you can add a custom dependency specification to the [dev-dependencies] section.

The [addresses] section is used to add aliases for the addresses. Any address can be specified in this section, and then used in the code as an alias. For example, if you add alice = "0xA11CE" to this section, you can use alice as 0xA11CE in the code.

The [dev-addresses] section is the same as [addresses] , but only works for the test and dev modes. Important to note that it is impossible to introduce new aliases in this section, only override the existing ones. So in the example above, if you add alice = "0xB0B" to this section, the alice address will be 0xB0B in the test and dev modes, and 0xA11CE in the regular build.

The TOML format supports two styles for tables: inline and multiline. The examples above are using the inline style, but it is also possible to use the multiline style. You wouldn't want to use it for the [package] section, but it can be useful for the dependencies.

```
```bash
```

## Inline style

```
[dependencies] Example = { override = true, git = "https://github.com/example/example.git", subdir = "crates/sui-  
framework/packages/sui-framework", rev = "framework/testnet" } MyPackage = { local = "../my-package" } ```
```

```
```bash
```

## Multiline style

```
[dependencies.Example] override = true git = "https://github.com/example/example.git" subdir = "crates/sui-  
framework/packages/sui-framework" rev = "framework/testnet"
```

```
[dependencies.MyPackage] local = "../my-package" ```
```

## TOML styles

The TOML format supports two styles for tables: inline and multiline. The examples above are using the inline style, but it is also possible to use the multiline style. You wouldn't want to use it for the [package] section, but it can be useful for the dependencies.

```
```bash
```

## Inline style

```
[dependencies] Example = { override = true, git = "https://github.com/example/example.git", subdir = "crates/sui-  
framework/packages/sui-framework", rev = "framework/testnet" } MyPackage = { local = "../my-package" } ```
```

```
```bash
```

## Multiline style

```
[dependencies.Example] override = true git = "https://github.com/example/example.git" subdir = "crates/sui-  
framework/packages/sui-framework" rev = "framework/testnet"
```

```
[dependencies.MyPackage] local = "../my-package"``
```

## **Further Reading**