The Move Book

Move has a unique type system which allows customizing type abilities . In the previous section , we introduced the struct definition and how to use it. However, the instances of the Artist and Record structs had to be unpacked for the code to compile. This is default behavior of a struct without abilities .

Throughout the book you will see chapters with name Ability: , where is the name of the ability. These chapters will cover the ability in detail, how it works, and how to use it in Move.

Abilities are a way to allow certain behaviors for a type. They are a part of the struct declaration and define which behaviors are allowed for the instances of the struct.

Abilities are set in the struct definition using the has keyword followed by a list of abilities. The abilities are separated by commas. Move supports 4 abilities: copy , drop , key , and store . Each ability defines a specific behavior for the struct instances.

A quick overview of the abilities:

All of the built-in types except references have copy , drop and store abilities. References have copy and drop .

While it is important to briefly mention them here, we will go into more detail about each ability in the following chapters and give proper context on how to use them.

A struct without abilities cannot be discarded, copied, or stored in storage. We call such a struct a Hot Potato . It is a joke, but it is also a good way to remember that a struct without abilities is like a hot potato - it can only be passed around and requires special handling. The Hot Potato is one of the most powerful patterns in Move, and we go into more detail about it in the Hot Potato Pattern chapter.

What are Abilities?

Abilities are a way to allow certain behaviors for a type. They are a part of the struct declaration and define which behaviors are allowed for the instances of the struct.

Abilities are set in the struct definition using the has keyword followed by a list of abilities. The abilities are separated by commas. Move supports 4 abilities: copy , drop , key , and store . Each ability defines a specific behavior for the struct instances.

```
bash /// This struct has the `copy` and `drop` abilities. struct VeryAble has copy, drop { //
field: Type1, // field2: Type2, // ... }
```

A quick overview of the abilities:

All of the built-in types except references have copy , drop and store abilities. References have copy and drop .

While it is important to briefly mention them here, we will go into more detail about each ability in the following chapters and give proper context on how to use them.

A struct without abilities cannot be discarded, copied, or stored in storage. We call such a struct a Hot Potato . It is a joke, but it is also a good way to remember that a struct without abilities is like a hot potato - it can only be passed around and requires special handling. The Hot Potato is one of the most powerful patterns in Move, and we go into more detail about it in the Hot Potato Pattern chapter.

Abilities syntax

Abilities are set in the struct definition using the has keyword followed by a list of abilities. The abilities are separated by commas. Move supports 4 abilities: copy , drop , key , and store . Each ability defines a specific behavior for the struct instances.

```
bash /// This struct has the `copy` and `drop` abilities. struct VeryAble has copy, drop { //
field: Type1, // field2: Type2, // ... }
```

A quick overview of the abilities:

All of the built-in types except references have copy , drop and store abilities. References have copy and drop .

While it is important to briefly mention them here, we will go into more detail about each ability in the following chapters and give proper context on how to use them.

A struct without abilities cannot be discarded, copied, or stored in storage. We call such a struct a Hot Potato . It is a joke, but it is also a good way to remember that a struct without abilities is like a hot potato - it can only be passed around and requires special handling. The Hot Potato is one of the most powerful patterns in Move, and we go into more detail about it in the Hot Potato Pattern chapter.

Overview

A quick overview of the abilities:

All of the built-in types except references have copy , drop and store abilities. References have copy and drop .

While it is important to briefly mention them here, we will go into more detail about each ability in the following chapters and give proper context on how to use them.

A struct without abilities cannot be discarded, copied, or stored in storage. We call such a struct a Hot Potato . It is a joke, but it is also a good way to remember that a struct without abilities is like a hot potato - it can only be passed around and requires special handling. The Hot Potato is one of the most powerful patterns in Move, and we go into more detail about it in the Hot Potato Pattern chapter.

No abilities

A struct without abilities cannot be discarded, copied, or stored in storage. We call such a struct a Hot Potato . It is a joke, but it is also a good way to remember that a struct without abilities is like a hot potato - it can only be passed around and requires special handling. The Hot Potato is one of the most powerful patterns in Move, and we go into more detail about it in the Hot Potato Pattern chapter.

Further reading