Custom Transfer Rules

Every Sui object must have the key ability. The store ability, on the other hand, is an optional ability you can add to Sui objects. Objects with the store ability:

Importantly for custom transfer rules, if the Sui object Object does not have the store ability, you cannot call the sui::transfer::public_transfer function to transfer it. The Move module that defines Object is the only entity that can transfer objects of that type using the sui::transfer::transfer function. Consequently, the module that defines the object Object can define a custom transfer function for Object that can take any number of arguments, and enforce any restrictions desired for performing a transfer operation (for example, a fee must be paid in order to transfer the object).

Custom transfer rules for objects enable you to define the transfer conditions that must be met for a valid transfer operation. You should be intentional about adding the store ability to an object because you are providing unrestricted access to that object without having to go through the module that defines it. After you enable public transfers on an object, there is no way of re-enabling custom transfer rules or any type of restrictions regarding the transfer of the object.

This example creates an object type Object that is transferrable only if the unlocked flag inside of it is set to true :

Within the same module that defines the object Object , you can then define a custom transfer rule transfer_unlocked for Object that takes the object to transfer and the address to transfer it to, and verifies that the object is unlocked before transferring it to the specified address.

With custom transfer rules, you can define multiple different transfer rules for the same object. Each of these rules might have different restrictions that execution of the transaction can dynamically enforce. So, if you wanted to allow only locked objects to be transferred to a specific address you could add the following function to the previous module:

With these rules in place, there are two different custom transfer rules for any object Object : either it's unlocked and anyone can transfer it, or it's locked, and it can only be transferred to 0xCAFE . Importantly, these two ways of transferring Object are the only ways of transferring any object of type Object . In particular, because Object does not have the store ability, you cannot transfer it using the sui::transfer::public_transfer function. In fact, the only ways of transferring Object are using examples::custom_transfer::transfer_unlocked and examples::custom_transfer::transfer_locked .

The store ability and transfer rules

Custom transfer rules for objects enable you to define the transfer conditions that must be met for a valid transfer operation. You should be intentional about adding the store ability to an object because you are providing unrestricted access to that object without having to go through the module that defines it. After you enable public transfers on an object, there is no way of re-enabling custom transfer rules or any type of restrictions regarding the transfer of the object.

This example creates an object type Object that is transferrable only if the unlocked flag inside of it is set to true :

Within the same module that defines the object Object , you can then define a custom transfer rule transfer_unlocked for Object that takes the object to transfer and the address to transfer it to, and verifies that the object is unlocked before transferring it to the specified address.

With custom transfer rules, you can define multiple different transfer rules for the same object. Each of these rules might have different restrictions that execution of the transaction can dynamically enforce. So, if you wanted to allow only locked objects to be transferred to a specific address you could add the following function to the previous module:

With these rules in place, there are two different custom transfer rules for any object Object : either it's unlocked and anyone can transfer it, or it's locked, and it can only be transferred to 0xCAFE . Importantly, these two ways of transferring Object are the only ways of transferring any object of type Object . In particular, because Object does not have the store ability, you cannot transfer it using the sui::transfer::public_transfer function. In fact, the only ways of transferring Object are using examples::custom_transfer::transfer_unlocked and examples::custom_transfer::transfer_locked .

Example

This example creates an object type Object that is transferrable only if the unlocked flag inside of it is set to true :

Within the same module that defines the object Object , you can then define a custom transfer rule transfer_unlocked for Object that takes the object to transfer and the address to transfer it to, and verifies that the object is unlocked before transferring it to the specified address.

With custom transfer rules, you can define multiple different transfer rules for the same object. Each of these rules might have different restrictions that execution of the transaction can dynamically enforce. So, if you wanted to allow only locked objects to be transferred to a specific address you could add the following function to the previous module:

With these rules in place, there are two different custom transfer rules for any object Object : either it's unlocked and anyone can transfer it, or it's locked, and it can only be transferred to 0xCAFE . Importantly, these two ways of transferring Object are the only ways of transferring any object of type Object . In particular, because Object does not have the store ability, you cannot transfer it using the sui::transfer::public_transfer function. In fact, the only ways of transferring Object are using examples::custom_transfer::transfer_unlocked and examples::custom_transfer::transfer_locked .