Module sui::coin

Defines the Coin type - platform wide representation of fungible tokens and coins. Coin can be described as a secure wrapper around Balance type.

A coin of type T worth value . Transferable and storable

Each Coin type T created through create_currency function will have a unique instance of CoinMetadata that stores the metadata for this coin type.

Similar to CoinMetadata, but created only for regulated coins that use the DenyList. This object is always immutable.

Capability allowing the bearer to mint and burn coins of type T. Transferable

Capability allowing the bearer to deny addresses from using the currency's coins-- immediately preventing those addresses from interacting with the coin as an input to a transaction and at the start of the next preventing them from receiving the coin. If allow_global_pause is true, the bearer can enable a global pause that behaves as if all addresses were added to the deny list.

Capability allowing the bearer to freeze addresses, preventing those addresses from interacting with the coin as an input to a transaction.

A type passed to create_supply is not a one-time witness.

Invalid arguments are passed to a function.

Trying to split a coin more times than its balance allows.

The index into the deny list vector for the sui::coin::Coin type.

Return the total number of T's in circulation.

Unwrap TreasuryCap getting the Supply.

Operation is irreversible. Supply cannot be converted into a TreasuryCap due to different security guarantees (TreasuryCap can be created only once for a type)

Get immutable reference to the treasury's Supply.

Get mutable reference to the treasury's Supply.

Public getter for the coin's value

Get immutable reference to the balance of a coin.

Get a mutable reference to the balance of a coin.

Wrap a balance into a Coin to make it transferable.

Destruct a Coin wrapper and keep the balance.

Take a Coin worth of value from Balance. Aborts if value > balance . value

Put a Coin to the Balance.

Consume the coin c and add its value to self. Aborts if c. value + self. value > U64_MAX

Split coin self to two coins, one with balance split_amount, and the remaining balance is left is self.

Split coin self into n - 1 coins with equal balances. The remainder is left in self. Return newly created coins.

Make any Coin with a zero value. Useful for placeholding bids/payments or preemptively making empty balances.

Destroy a coin with value zero

Create a new currency type T as and return the TreasuryCap for T to the caller. Can only be called with a one-time-witness type,

ensuring that there's only one TreasuryCap per T.

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. When an address is added to the deny list, it is immediately unable to interact with the currency's coin as input objects. Additionally at the start of the next epoch, they will be unable to receive the currency's coin. The allow_global_pause flag enables an additional API that will cause all addresses to be denied. Note however, that this doesn't affect per-address entries of the deny list and will not change the result of the "contains" APIs.

Given the DenyCap for a regulated currency, migrate it to the new DenyCapV2 type. All entries in the deny list will be migrated to the new format. See create_regulated_currency_v2 for details on the new v2 of the deny list.

Create a coin worth value and increase the total supply in cap accordingly.

Mint some amount of T as a Balance and increase the total supply in cap accordingly. Aborts if value + cap. total_supply >= U64_MAX

Destroy the coin c and decrease the total supply in cap accordingly.

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

Check if the global pause is enabled for the given coin type in the current epoch.

Check if the global pause is enabled for the given coin type in the next epoch.

Mint amount of Coin and send it to recipient. Invokes mint ().

Update name of the coin in CoinMetadata

Update the symbol of the coin in CoinMetadata

Update the description of the coin in CoinMetadata

Update the url of the coin in CoinMetadata

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

Struct

A coin of type T worth value . Transferable and storable

```bash
```

```

Each Coin type T created through create_currency function will have a unique instance of CoinMetadata that stores the metadata for this coin type.

```bash

```

Similar to CoinMetadata, but created only for regulated coins that use the DenyList. This object is always immutable.

```bash

```

Capability allowing the bearer to mint and burn coins of type T. Transferable

```bash

```

Capability allowing the bearer to deny addresses from using the currency's coins-- immediately preventing those addresses from interacting with the coin as an input to a transaction and at the start of the next preventing them from receiving the coin. If allow_global_pause is true, the bearer can enable a global pause that behaves as if all addresses were added to the deny list.

```bash

```

```bash

```

Capability allowing the bearer to freeze addresses, preventing those addresses from interacting with the coin as an input to a transaction.

```bash

```

A type passed to create_supply is not a one-time witness.

```bash

```

Invalid arguments are passed to a function.

```bash

```

Trying to split a coin more times than its balance allows.

```bash

```

```bash

```

The index into the deny list vector for the sui::coin::Coin type.

```bash

```
```

Return the total number of T's in circulation.

```bash
```

```bash
```

Unwrap TreasuryCap getting the Supply.

Operation is irreversible. Supply cannot be converted into a TreasuryCap due to different security guarantees (TreasuryCap can be created only once for a type)

```bash
```

```bash
```

Get immutable reference to the treasury's Supply.

```bash
```

```bash
```

Get mutable reference to the treasury's Supply.

```bash
```

```bash
```

Public getter for the coin's value

```bash
```

```bash
```

Get immutable reference to the balance of a coin.

```bash
```

```bash
```

Get a mutable reference to the balance of a coin.

```bash
```

```

```bash
```

Wrap a balance into a Coin to make it transferable.

```bash
```

```bash
```

Destruct a Coin wrapper and keep the balance.

```bash
```

```bash
```

Take a Coin worth of value from Balance. Aborts if value > balance . value

```bash
```

```bash
```

Put a Coin to the Balance.

```bash
```

```bash
```

Consume the coin c and add its value to self. Aborts if c. value + self. value > U64_MAX

```bash
```

```bash
```

Split coin self to two coins, one with balance split_amount, and the remaining balance is left is self.

```bash
```

```bash
```

Split coin self into n - 1 coins with equal balances. The remainder is left in self. Return newly created coins.
```

```bash
```

```bash
```

Make any Coin with a zero value. Useful for placeholding bids/payments or preemptively making empty balances.

```bash
```

```bash
```

Destroy a coin with value zero

```bash
```

```bash
```

Create a new currency type T as and return the TreasuryCap for T to the caller. Can only be called with a one-time-witness type, ensuring that there's only one TreasuryCap per T.

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. When an address is added to the deny list, it is immediately unable to interact with the currency's coin as input objects. Additionally at the start of the next epoch, they will be unable to receive the currency's coin. The allow_global_pause flag enables an additional API that will cause all addresses to be denied. Note however, that this doesn't affect per-address entries of the deny list and will not change the result of the "contains" APIs.

```bash
```

```bash
```

Given the DenyCap for a regulated currency, migrate it to the new DenyCapV2 type. All entries in the deny list will be migrated to the new format. See create_regulated_currency_v2 for details on the new v2 of the deny list.

```bash
```

```bash
```

Create a coin worth value and increase the total supply in cap accordingly.

```bash
```

```

```bash
```

Mint some amount of T as a Balance and increase the total supply in cap accordingly. Aborts if value + cap. total_supply >= U64_MAX

```bash
```

```bash
```

Destroy the coin c and decrease the total supply in cap accordingly.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash
```

```bash
```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash
```

```bash
```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```
```

```bash
```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash
```

```bash
```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash
```

```bash
```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash
```

```bash
```

Update name of the coin in CoinMetadata

```bash
```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Struct

Each Coin type T created through create_currency function will have a unique instance of CoinMetadata that stores the metadata for this coin type.

```bash
```

Similar to CoinMetadata, but created only for regulated coins that use the DenyList. This object is always immutable.

```bash
```

Capability allowing the bearer to mint and burn coins of type T. Transferable

```bash

```

Capability allowing the bearer to deny addresses from using the currency's coins-- immediately preventing those addresses from interacting with the coin as an input to a transaction and at the start of the next preventing them from receiving the coin. If allow_global_pause is true, the bearer can enable a global pause that behaves as if all addresses were added to the deny list.

```bash

```

```bash

```

Capability allowing the bearer to freeze addresses, preventing those addresses from interacting with the coin as an input to a transaction.

```bash

```

A type passed to create_supply is not a one-time witness.

```bash

```

Invalid arguments are passed to a function.

```bash

```

Trying to split a coin more times than its balance allows.

```bash

```

```bash

```

The index into the deny list vector for the sui::coin::Coin type.

```bash

```

Return the total number of T's in circulation.

```bash

```

```bash

```

Unwrap TreasuryCap getting the Supply.

Operation is irreversible. Supply cannot be converted into a TreasuryCap due to different security guarantees (TreasuryCap can be created only once for a type)

```bash
```

```

```bash
```

Get immutable reference to the treasury's Supply.

```bash
```

```bash
```

Get mutable reference to the treasury's Supply.

```bash
```

```bash
```

Public getter for the coin's value

```bash
```

```bash
```

Get immutable reference to the balance of a coin.

```bash
```

```bash
```

Get a mutable reference to the balance of a coin.

```bash
```

```bash
```

Wrap a balance into a Coin to make it transferable.

```bash
```

```bash
```

Destruct a Coin wrapper and keep the balance.
```

```bash
```

```bash
```

Take a Coin worth of value from Balance. Aborts if value > balance . value

```bash
```

```bash
```

Put a Coin to the Balance.

```bash
```

```bash
```

Consume the coin c and add its value to self. Aborts if c. value + self. value > U64_MAX

```bash
```

```bash
```

Split coin self to two coins, one with balance split_amount, and the remaining balance is left is self.

```bash
```

```bash
```

Split coin self into n - 1 coins with equal balances. The remainder is left in self. Return newly created coins.

```bash
```

```bash
```

Make any Coin with a zero value. Useful for placeholding bids/payments or preemptively making empty balances.

```bash
```

```bash
```

Destroy a coin with value zero

```bash
```

```bash
```

Create a new currency type T as and return the TreasuryCap for T to the caller. Can only be called with a one-time-witness type, ensuring that there's only one TreasuryCap per T.

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. When an address is added to the deny list, it is immediately unable to interact with the currency's coin as input objects. Additionally at the start of the next epoch, they will be unable to receive the currency's coin. The allow_global_pause flag enables an additional API that will cause all addresses to be denied. Note however, that this doesn't affect per-address entries of the deny list and will not change the result of the "contains" APIs.

```bash
```

```bash
```

Given the DenyCap for a regulated currency, migrate it to the new DenyCapV2 type. All entries in the deny list will be migrated to the new format. See create_regulated_currency_v2 for details on the new v2 of the deny list.

```bash
```

```bash
```

Create a coin worth value and increase the total supply in cap accordingly.

```bash
```

```bash
```

Mint some amount of T as a Balance and increase the total supply in cap accordingly. Aborts if value + cap. total_supply >= U64_MAX

```bash
```

```bash
```

Destroy the coin c and decrease the total supply in cap accordingly.

```bash

```

```bash

```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash

```

```bash

```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash

```

```bash

```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash

```

```bash

```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash

```

```bash

```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash

```

```bash

```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as

inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash
```

```bash
```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash
```

```bash
```

Update name of the coin in CoinMetadata

```bash
```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Struct

Similar to CoinMetadata, but created only for regulated coins that use the DenyList. This object is always immutable.

```bash
```

Capability allowing the bearer to mint and burn coins of type T. Transferable

```bash
```

Capability allowing the bearer to deny addresses from using the currency's coins-- immediately preventing those addresses from interacting with the coin as an input to a transaction and at the start of the next preventing them from receiving the coin. If allow_global_pause is true, the bearer can enable a global pause that behaves as if all addresses were added to the deny list.

```bash
```

```bash
```

Capability allowing the bearer to freeze addresses, preventing those addresses from interacting with the coin as an input to a transaction.

```bash
```

A type passed to create_supply is not a one-time witness.

```bash
```

Invalid arguments are passed to a function.

```bash
```

Trying to split a coin more times than its balance allows.

```bash
```

```bash
```

The index into the deny list vector for the sui::coin::Coin type.

```bash
```

Return the total number of T's in circulation.

```bash
```

```bash
```

Unwrap TreasuryCap getting the Supply.

Operation is irreversible. Supply cannot be converted into a TreasuryCap due to different security guarantees (TreasuryCap can be created only once for a type)

```bash
```

```bash
```

Get immutable reference to the treasury's Supply.

```bash
```

```bash
```

Get mutable reference to the treasury's Supply.

```bash
```

```bash
```

Public getter for the coin's value

```bash
```

```bash
```

Get immutable reference to the balance of a coin.

```bash
```

```bash
```

Get a mutable reference to the balance of a coin.

```bash
```

```bash
```

Wrap a balance into a Coin to make it transferable.

```bash
```

```bash
```

Destruct a Coin wrapper and keep the balance.

```bash
```

```bash
```

Take a Coin worth of value from Balance. Aborts if value > balance . value

```bash
```

```bash
```

Put a Coin to the Balance.

```bash
```

```

```bash
```

Consume the coin c and add its value to self. Aborts if c. value + self. value > U64_MAX

```bash
```

```bash
```

Split coin self to two coins, one with balance split_amount, and the remaining balance is left is self.

```bash
```

```bash
```

Split coin self into n - 1 coins with equal balances. The remainder is left in self. Return newly created coins.

```bash
```

```bash
```

Make any Coin with a zero value. Useful for placeholding bids/payments or preemptively making empty balances.

```bash
```

```bash
```

Destroy a coin with value zero

```bash
```

```bash
```

Create a new currency type T as and return the TreasuryCap for T to the caller. Can only be called with a one-time-witness type, ensuring that there's only one TreasuryCap per T.

```bash
```

```bash
```
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. When an address is added to the deny list, it is immediately unable to interact with the currency's coin as input objects. Additionally at the start of the next epoch, they will be unable to receive the currency's coin. The allow_global_pause flag enables an additional API that will cause all addresses to be denied. Note however, that this doesn't affect per-address entries of the deny list and will not change the result of the "contains" APIs.

```bash
```

```bash
```

Given the DenyCap for a regulated currency, migrate it to the new DenyCapV2 type. All entries in the deny list will be migrated to the new format. See create_regulated_currency_v2 for details on the new v2 of the deny list.

```bash
```

```bash
```

Create a coin worth value and increase the total supply in cap accordingly.

```bash
```

```bash
```

Mint some amount of T as a Balance and increase the total supply in cap accordingly. Aborts if value + cap. total_supply >= U64_MAX

```bash
```

```bash
```

Destroy the coin c and decrease the total supply in cap accordingly.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash
```

```bash
```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash
```

```bash
```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash
```

```bash
```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash
```

```bash
```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash
```

```bash
```

Update name of the coin in CoinMetadata

```bash
```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```
```

```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash

```

```bash

```

Struct

Capability allowing the bearer to mint and burn coins of type T. Transferable

```bash

```

Capability allowing the bearer to deny addresses from using the currency's coins-- immediately preventing those addresses from interacting with the coin as an input to a transaction and at the start of the next preventing them from receiving the coin. If allow_global_pause is true, the bearer can enable a global pause that behaves as if all addresses were added to the deny list.

```bash

```

```bash

```

Capability allowing the bearer to freeze addresses, preventing those addresses from interacting with the coin as an input to a transaction.

```bash

```

A type passed to create_supply is not a one-time witness.

```bash

```

Invalid arguments are passed to a function.

```bash

```

Trying to split a coin more times than its balance allows.

```bash

```

```bash

```

The index into the deny list vector for the sui::coin::Coin type.

```bash

```
```

Return the total number of T's in circulation.

```bash
```

```bash
```

Unwrap TreasuryCap getting the Supply.

Operation is irreversible. Supply cannot be converted into a TreasuryCap due to different security guarantees (TreasuryCap can be created only once for a type)

```bash
```

```bash
```

Get immutable reference to the treasury's Supply.

```bash
```

```bash
```

Get mutable reference to the treasury's Supply.

```bash
```

```bash
```

Public getter for the coin's value

```bash
```

```bash
```

Get immutable reference to the balance of a coin.

```bash
```

```bash
```

Get a mutable reference to the balance of a coin.

```bash
```

```

```bash
```

Wrap a balance into a Coin to make it transferable.

```bash
```

```bash
```

Destruct a Coin wrapper and keep the balance.

```bash
```

```bash
```

Take a Coin worth of value from Balance. Aborts if value > balance . value

```bash
```

```bash
```

Put a Coin to the Balance.

```bash
```

```bash
```

Consume the coin c and add its value to self. Aborts if c. value + self. value > U64_MAX

```bash
```

```bash
```

Split coin self to two coins, one with balance split_amount, and the remaining balance is left is self.

```bash
```

```bash
```

Split coin self into n - 1 coins with equal balances. The remainder is left in self. Return newly created coins.
```

```bash
```

```bash
```

Make any Coin with a zero value. Useful for placeholding bids/payments or preemptively making empty balances.

```bash
```

```bash
```

Destroy a coin with value zero

```bash
```

```bash
```

Create a new currency type T as and return the TreasuryCap for T to the caller. Can only be called with a one-time-witness type, ensuring that there's only one TreasuryCap per T.

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. When an address is added to the deny list, it is immediately unable to interact with the currency's coin as input objects. Additionally at the start of the next epoch, they will be unable to receive the currency's coin. The allow_global_pause flag enables an additional API that will cause all addresses to be denied. Note however, that this doesn't affect per-address entries of the deny list and will not change the result of the "contains" APIs.

```bash
```

```bash
```

Given the DenyCap for a regulated currency, migrate it to the new DenyCapV2 type. All entries in the deny list will be migrated to the new format. See create_regulated_currency_v2 for details on the new v2 of the deny list.

```bash
```

```bash
```

Create a coin worth value and increase the total supply in cap accordingly.

```bash
```

```

```bash
```

Mint some amount of T as a Balance and increase the total supply in cap accordingly. Aborts if value + cap. total_supply >= U64_MAX

```bash
```

```bash
```

Destroy the coin c and decrease the total supply in cap accordingly.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash
```

```bash
```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash
```

```bash
```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```
```

```bash
```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash
```

```bash
```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash
```

```bash
```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash
```

```bash
```

Update name of the coin in CoinMetadata

```bash
```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Struct

Capability allowing the bearer to deny addresses from using the currency's coins-- immediately preventing those addresses from interacting with the coin as an input to a transaction and at the start of the next preventing them from receiving the coin. If allow_global_pause is true, the bearer can enable a global pause that behaves as if all addresses were added to the deny list.

```bash
```

```bash
```

Capability allowing the bearer to freeze addresses, preventing those addresses from interacting with the coin as an input to a transaction.

```bash
```

A type passed to create_supply is not a one-time witness.

```bash
```

Invalid arguments are passed to a function.

```bash
```

Trying to split a coin more times than its balance allows.

```bash
```

```bash
```

The index into the deny list vector for the sui::coin::Coin type.

```bash
```

Return the total number of T's in circulation.

```bash
```

```bash
```

Unwrap TreasuryCap getting the Supply.

Operation is irreversible. Supply cannot be converted into a TreasuryCap due to different security guarantees (TreasuryCap can be created only once for a type)

```bash
```

```bash
```

Get immutable reference to the treasury's Supply.

```bash
```

```bash
```

Get mutable reference to the treasury's Supply.

```bash
```

```bash
```

Public getter for the coin's value

```bash
```

```bash
```

Get immutable reference to the balance of a coin.

```bash
```

```bash
```

Get a mutable reference to the balance of a coin.

```bash
```

```bash
```

Wrap a balance into a Coin to make it transferable.

```bash
```

```bash
```

Destruct a Coin wrapper and keep the balance.

```bash
```

```bash
```

Take a Coin worth of value from Balance. Aborts if value > balance . value

```bash
```

```bash
```

Put a Coin to the Balance.

```bash
```

```bash
```

Consume the coin c and add its value to self. Aborts if c. value + self. value > U64_MAX

```bash
```

```bash
```

Split coin self to two coins, one with balance split_amount, and the remaining balance is left is self.

```bash
```

```bash
```

Split coin self into n - 1 coins with equal balances. The remainder is left in self. Return newly created coins.

```bash
```

```bash
```

Make any Coin with a zero value. Useful for placeholding bids/payments or preemptively making empty balances.

```bash
```

```bash
```

Destroy a coin with value zero

```bash
```

```bash
```

Create a new currency type T as and return the TreasuryCap for T to the caller. Can only be called with a one-time-witness type, ensuring that there's only one TreasuryCap per T.

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. When an address is added to the deny list, it is immediately unable to interact with the currency's coin as input objects. Additionally at the start of the next epoch, they will be unable to receive the currency's coin. The allow_global_pause flag enables an additional API that will cause all addresses to be denied. Note however, that this doesn't affect per-address entries of the deny list and will not change the result of the "contains" APIs.

```bash
```

```bash
```

Given the DenyCap for a regulated currency, migrate it to the new DenyCapV2 type. All entries in the deny list will be migrated to the new format. See create_regulated_currency_v2 for details on the new v2 of the deny list.

```bash
```

```bash
```

Create a coin worth value and increase the total supply in cap accordingly.

```bash
```

```bash
```

Mint some amount of T as a Balance and increase the total supply in cap accordingly. Aborts if value + cap. total_supply >= U64_MAX

```bash
```

```bash
```

Destroy the coin c and decrease the total supply in cap accordingly.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash
```

```bash
```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash
```

```bash
```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash
```

```bash
```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash
```

```bash
```

```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash

```

```bash

```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash

```

```bash

```

Update name of the coin in CoinMetadata

```bash

```

```bash

```

Update the symbol of the coin in CoinMetadata

```bash

```

```bash

```

Update the description of the coin in CoinMetadata

```bash

```

```bash

```

Update the url of the coin in CoinMetadata

```bash

```

```bash

```

```bash

```

```bash

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```
```

```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Struct

```bash
```

Capability allowing the bearer to freeze addresses, preventing those addresses from interacting with the coin as an input to a transaction.

```bash
```

A type passed to create_supply is not a one-time witness.

```bash
```

Invalid arguments are passed to a function.

```bash
```

Trying to split a coin more times than its balance allows.

```bash
```

```bash
```

The index into the deny list vector for the sui::coin::Coin type.

```bash
```

Return the total number of T's in circulation.

```bash
```

```bash
```
```

Unwrap TreasuryCap getting the Supply.

Operation is irreversible. Supply cannot be converted into a TreasuryCap due to different security guarantees (TreasuryCap can be created only once for a type)

```bash

```

```bash

```

Get immutable reference to the treasury's Supply.

```bash

```

```bash

```

Get mutable reference to the treasury's Supply.

```bash

```

```bash

```

Public getter for the coin's value

```bash

```

```bash

```

Get immutable reference to the balance of a coin.

```bash

```

```bash

```

Get a mutable reference to the balance of a coin.

```bash

```

```bash

```

Wrap a balance into a Coin to make it transferable.

```bash
```

```

```bash
```

Destruct a Coin wrapper and keep the balance.

```bash
```

```bash
```

Take a Coin worth of value from Balance. Aborts if value > balance . value

```bash
```

```bash
```

Put a Coin to the Balance.

```bash
```

```bash
```

Consume the coin c and add its value to self. Aborts if c. value + self. value > U64_MAX

```bash
```

```bash
```

Split coin self to two coins, one with balance split_amount, and the remaining balance is left is self.

```bash
```

```bash
```

Split coin self into n - 1 coins with equal balances. The remainder is left in self. Return newly created coins.

```bash
```

```bash
```

Make any Coin with a zero value. Useful for placeholding bids/payments or preemptively making empty balances.
```

```bash
```

```bash
```

Destroy a coin with value zero

```bash
```

```bash
```

Create a new currency type T as and return the TreasuryCap for T to the caller. Can only be called with a one-time-witness type, ensuring that there's only one TreasuryCap per T.

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. When an address is added to the deny list, it is immediately unable to interact with the currency's coin as input objects. Additionally at the start of the next epoch, they will be unable to receive the currency's coin. The allow_global_pause flag enables an additional API that will cause all addresses to be denied. Note however, that this doesn't affect per-address entries of the deny list and will not change the result of the "contains" APIs.

```bash
```

```bash
```

Given the DenyCap for a regulated currency, migrate it to the new DenyCapV2 type. All entries in the deny list will be migrated to the new format. See create_regulated_currency_v2 for details on the new v2 of the deny list.

```bash
```

```bash
```

Create a coin worth value and increase the total supply in cap accordingly.

```bash
```

```bash
```

Mint some amount of T as a Balance and increase the total supply in cap accordingly. Aborts if value + cap. total_supply >= U64_MAX

```bash
```

```bash
```

Destroy the coin c and decrease the total supply in cap accordingly.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash
```

```bash
```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash
```

```bash
```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash
```

```

```bash
```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash
```

```bash
```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash
```

```bash
```

Update name of the coin in CoinMetadata

```bash
```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Struct

Capability allowing the bearer to freeze addresses, preventing those addresses from interacting with the coin as an input to a transaction.

```bash
```

A type passed to create_supply is not a one-time witness.

```bash
```

Invalid arguments are passed to a function.

```bash
```

Trying to split a coin more times than its balance allows.

```bash
```

```

```bash
```

The index into the deny list vector for the sui::coin::Coin type.

```bash
```

Return the total number of T's in circulation.

```bash
```

```bash
```

Unwrap TreasuryCap getting the Supply.

Operation is irreversible. Supply cannot be converted into a TreasuryCap due to different security guarantees (TreasuryCap can be created only once for a type)

```bash
```

```bash
```

Get immutable reference to the treasury's Supply.

```bash
```

```bash
```

Get mutable reference to the treasury's Supply.

```bash
```

```bash
```

Public getter for the coin's value

```bash
```

```bash
```

Get immutable reference to the balance of a coin.
```

```bash
```

```bash
```

Get a mutable reference to the balance of a coin.

```bash
```

```bash
```

Wrap a balance into a Coin to make it transferable.

```bash
```

```bash
```

Destruct a Coin wrapper and keep the balance.

```bash
```

```bash
```

Take a Coin worth of value from Balance. Aborts if value > balance . value

```bash
```

```bash
```

Put a Coin to the Balance.

```bash
```

```bash
```

Consume the coin c and add its value to self. Aborts if c. value + self. value > U64_MAX

```bash
```

```bash
```

Split coin self to two coins, one with balance split_amount, and the remaining balance is left is self.

```bash
```

```bash
```

Split coin self into n - 1 coins with equal balances. The remainder is left in self. Return newly created coins.

```bash
```

```bash
```

Make any Coin with a zero value. Useful for placeholding bids/payments or preemptively making empty balances.

```bash
```

```bash
```

Destroy a coin with value zero

```bash
```

```bash
```

Create a new currency type T as and return the TreasuryCap for T to the caller. Can only be called with a one-time-witness type, ensuring that there's only one TreasuryCap per T.

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. When an address is added to the deny list, it is immediately unable to interact with the currency's coin as input objects. Additionally at the start of the next epoch, they will be unable to receive the currency's coin. The allow_global_pause flag enables an additional API that will cause all addresses to be denied. Note however, that this doesn't affect per-address entries of the deny list and will not change the result of the "contains" APIs.

```bash
```

```bash
```

Given the DenyCap for a regulated currency, migrate it to the new DenyCapV2 type. All entries in the deny list will be migrated to the new format. See create_regulated_currency_v2 for details on the new v2 of the deny list.

```bash
```

```bash
```

Create a coin worth value and increase the total supply in cap accordingly.

```bash
```

```bash
```

Mint some amount of T as a Balance and increase the total supply in cap accordingly. Aborts if value + cap. total_supply >= U64_MAX

```bash
```

```bash
```

Destroy the coin c and decrease the total supply in cap accordingly.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash
```

```bash
```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash
```

```bash
```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash
```

```bash
```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash
```

```bash
```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash
```

```bash
```

Update name of the coin in CoinMetadata

```bash
```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Constants

A type passed to create_supply is not a one-time witness.

```bash
```

```

Invalid arguments are passed to a function.

```bash

```

Trying to split a coin more times than its balance allows.

```bash

```

```bash

```

The index into the deny list vector for the sui::coin::Coin type.

```bash

```

Return the total number of T's in circulation.

```bash

```

```bash

```

Unwrap TreasuryCap getting the Supply.

Operation is irreversible. Supply cannot be converted into a TreasuryCap due to different security guarantees (TreasuryCap can be created only once for a type)

```bash

```

```bash

```

Get immutable reference to the treasury's Supply.

```bash

```

```bash

```

Get mutable reference to the treasury's Supply.

```bash

```

```bash

```
```

Public getter for the coin's value

```bash
```

```bash
```

Get immutable reference to the balance of a coin.

```bash
```

```bash
```

Get a mutable reference to the balance of a coin.

```bash
```

```bash
```

Wrap a balance into a Coin to make it transferable.

```bash
```

```bash
```

Destruct a Coin wrapper and keep the balance.

```bash
```

```bash
```

Take a Coin worth of value from Balance. Aborts if value > balance . value

```bash
```

```bash
```

Put a Coin to the Balance.

```bash
```

```bash
```

```

Consume the coin c and add its value to self. Aborts if c. value + self. value > U64_MAX

```bash

```

```bash

```

Split coin self to two coins, one with balance split_amount, and the remaining balance is left is self.

```bash

```

```bash

```

Split coin self into n - 1 coins with equal balances. The remainder is left in self. Return newly created coins.

```bash

```

```bash

```

Make any Coin with a zero value. Useful for placeholding bids/payments or preemptively making empty balances.

```bash

```

```bash

```

Destroy a coin with value zero

```bash

```

```bash

```

Create a new currency type T as and return the TreasuryCap for T to the caller. Can only be called with a one-time-witness type, ensuring that there's only one TreasuryCap per T.

```bash

```

```bash

```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. When an address is added to the deny list, it is immediately unable to interact with the currency's coin as input objects. Additionally at the start of the next epoch, they will be unable to receive the currency's coin. The allow_global_pause flag enables an additional API that will cause all addresses to be denied. Note however, that this doesn't affect per-address entries of the deny list
```

and will not change the result of the "contains" APIs.

```bash
```

```bash
```

Given the DenyCap for a regulated currency, migrate it to the new DenyCapV2 type. All entries in the deny list will be migrated to the new format. See create_regulated_currency_v2 for details on the new v2 of the deny list.

```bash
```

```bash
```

Create a coin worth value and increase the total supply in cap accordingly.

```bash
```

```bash
```

Mint some amount of T as a Balance and increase the total supply in cap accordingly. Aborts if value + cap. total_supply >= U64_MAX

```bash
```

```bash
```

Destroy the coin c and decrease the total supply in cap accordingly.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash
```

```

```bash

```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash

```

```bash

```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash

```

```bash

```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash

```

```bash

```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash

```

```bash

```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash

```

```bash

```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash

```
```

```bash
```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash
```

```bash
```

Update name of the coin in CoinMetadata

```bash
```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.
```

```bash
```

```bash
```

Function

Return the total number of T's in circulation.

```bash
```

```bash
```

Unwrap TreasuryCap getting the Supply.

Operation is irreversible. Supply cannot be converted into a TreasuryCap due to different security guarantees (TreasuryCap can be created only once for a type)

```bash
```

```bash
```

Get immutable reference to the treasury's Supply.

```bash
```

```bash
```

Get mutable reference to the treasury's Supply.

```bash
```

```bash
```

Public getter for the coin's value

```bash
```

```bash
```

Get immutable reference to the balance of a coin.

```bash
```

```

```bash
```

Get a mutable reference to the balance of a coin.

```bash
```

```bash
```

Wrap a balance into a Coin to make it transferable.

```bash
```

```bash
```

Destruct a Coin wrapper and keep the balance.

```bash
```

```bash
```

Take a Coin worth of value from Balance. Aborts if value > balance . value

```bash
```

```bash
```

Put a Coin to the Balance.

```bash
```

```bash
```

Consume the coin c and add its value to self. Aborts if c. value + self. value > U64_MAX

```bash
```

```bash
```

Split coin self to two coins, one with balance split_amount, and the remaining balance is left is self.
```

```bash
```

```bash
```

Split coin self into n - 1 coins with equal balances. The remainder is left in self. Return newly created coins.

```bash
```

```bash
```

Make any Coin with a zero value. Useful for placeholding bids/payments or preemptively making empty balances.

```bash
```

```bash
```

Destroy a coin with value zero

```bash
```

```bash
```

Create a new currency type T as and return the TreasuryCap for T to the caller. Can only be called with a one-time-witness type, ensuring that there's only one TreasuryCap per T.

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. When an address is added to the deny list, it is immediately unable to interact with the currency's coin as input objects. Additionally at the start of the next epoch, they will be unable to receive the currency's coin. The allow_global_pause flag enables an additional API that will cause all addresses to be denied. Note however, that this doesn't affect per-address entries of the deny list and will not change the result of the "contains" APIs.

```bash
```

```bash
```

Given the DenyCap for a regulated currency, migrate it to the new DenyCapV2 type. All entries in the deny list will be migrated to the new format. See create_regulated_currency_v2 for details on the new v2 of the deny list.

```bash
```

```

```bash
```

Create a coin worth value and increase the total supply in cap accordingly.

```bash
```

```bash
```

Mint some amount of T as a Balance and increase the total supply in cap accordingly. Aborts if value + cap. total_supply >= U64_MAX

```bash
```

```bash
```

Destroy the coin c and decrease the total supply in cap accordingly.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash
```

```bash
```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash
```

```bash
```
```

```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash

```

```bash

```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash

```

```bash

```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash

```

```bash

```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash

```

```bash

```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash

```

```bash

```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash

```

```bash

```

Update name of the coin in CoinMetadata
```

```bash
```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

Unwrap TreasuryCap getting the Supply.

Operation is irreversible. Supply cannot be converted into a TreasuryCap due to different security guarantees (TreasuryCap can be created only once for a type)
```

```bash
```

```bash
```

Get immutable reference to the treasury's Supply.

```bash
```

```bash
```

Get mutable reference to the treasury's Supply.

```bash
```

```bash
```

Public getter for the coin's value

```bash
```

```bash
```

Get immutable reference to the balance of a coin.

```bash
```

```bash
```

Get a mutable reference to the balance of a coin.

```bash
```

```bash
```

Wrap a balance into a Coin to make it transferable.

```bash
```

```bash
```

Destruct a Coin wrapper and keep the balance.

```bash
```

```bash
```

Take a Coin worth of value from Balance. Aborts if value > balance . value

```bash
```

```bash
```

Put a Coin to the Balance.

```bash
```

```bash
```

Consume the coin c and add its value to self. Aborts if c. value + self. value > U64_MAX

```bash
```

```bash
```

Split coin self to two coins, one with balance split_amount, and the remaining balance is left is self.

```bash
```

```bash
```

Split coin self into n - 1 coins with equal balances. The remainder is left in self. Return newly created coins.

```bash
```

```bash
```

Make any Coin with a zero value. Useful for placeholding bids/payments or preemptively making empty balances.

```bash
```

```bash
```

```

Destroy a coin with value zero

```bash

```

```bash

```

Create a new currency type T as and return the TreasuryCap for T to the caller. Can only be called with a one-time-witness type, ensuring that there's only one TreasuryCap per T.

```bash

```

```bash

```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. When an address is added to the deny list, it is immediately unable to interact with the currency's coin as input objects. Additionally at the start of the next epoch, they will be unable to receive the currency's coin. The allow_global_pause flag enables an additional API that will cause all addresses to be denied. Note however, that this doesn't affect per-address entries of the deny list and will not change the result of the "contains" APIs.

```bash

```

```bash

```

Given the DenyCap for a regulated currency, migrate it to the new DenyCapV2 type. All entries in the deny list will be migrated to the new format. See create_regulated_currency_v2 for details on the new v2 of the deny list.

```bash

```

```bash

```

Create a coin worth value and increase the total supply in cap accordingly.

```bash

```

```bash

```

Mint some amount of T as a Balance and increase the total supply in cap accordingly. Aborts if value + cap. total_supply >= U64_MAX

```bash

```

```bash

```

Destroy the coin c and decrease the total supply in cap accordingly.

```bash

```

```bash

```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash

```

```bash

```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash

```

```bash

```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash

```

```bash

```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash

```

```bash

```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash

```

```bash

```
```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash
```

```bash
```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash
```

```bash
```

Update name of the coin in CoinMetadata

```bash
```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

Get immutable reference to the treasury's Supply.

```bash
```

```bash
```

Get mutable reference to the treasury's Supply.

```bash
```

```bash
```

Public getter for the coin's value

```bash
```

```bash
```
```

Get immutable reference to the balance of a coin.

```bash
```

```bash
```

Get a mutable reference to the balance of a coin.

```bash
```

```bash
```

Wrap a balance into a Coin to make it transferable.

```bash
```

```bash
```

Destruct a Coin wrapper and keep the balance.

```bash
```

```bash
```

Take a Coin worth of value from Balance. Aborts if value > balance . value

```bash
```

```bash
```

Put a Coin to the Balance.

```bash
```

```bash
```

Consume the coin c and add its value to self. Aborts if c. value + self. value > U64_MAX

```bash
```

```bash
```

```

Split coin self to two coins, one with balance split_amount, and the remaining balance is left is self.

```bash

```

```bash

```

Split coin self into n - 1 coins with equal balances. The remainder is left in self. Return newly created coins.

```bash

```

```bash

```

Make any Coin with a zero value. Useful for placeholding bids/payments or preemptively making empty balances.

```bash

```

```bash

```

Destroy a coin with value zero

```bash

```

```bash

```

Create a new currency type T as and return the TreasuryCap for T to the caller. Can only be called with a one-time-witness type, ensuring that there's only one TreasuryCap per T.

```bash

```

```bash

```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. When an address is added to the deny list, it is immediately unable to interact with the currency's coin as input objects. Additionally at the start of the next epoch, they will be unable to receive the currency's coin. The allow_global_pause flag enables an additional API that will cause all addresses to be denied. Note however, that this doesn't affect per-address entries of the deny list and will not change the result of the "contains" APIs.

```bash

```

```bash

```
```

Given the DenyCap for a regulated currency, migrate it to the new DenyCapV2 type. All entries in the deny list will be migrated to the new format. See create_regulated_currency_v2 for details on the new v2 of the deny list.

```bash

```

```bash

```

Create a coin worth value and increase the total supply in cap accordingly.

```bash

```

```bash

```

Mint some amount of T as a Balance and increase the total supply in cap accordingly. Aborts if value + cap. total_supply >= U64_MAX

```bash

```

```bash

```

Destroy the coin c and decrease the total supply in cap accordingly.

```bash

```

```bash

```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash

```

```bash

```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash

```

```bash

```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash
```

```bash
```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash
```

```bash
```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash
```

```bash
```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash
```

```bash
```

Update name of the coin in CoinMetadata

```bash
```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

Get mutable reference to the treasury's Supply.

```bash
```

```bash
```

Public getter for the coin's value

```bash
```

```bash
```

Get immutable reference to the balance of a coin.

```bash
```

```bash
```

Get a mutable reference to the balance of a coin.

```bash
```

```bash
```

Wrap a balance into a Coin to make it transferable.

```bash
```

```bash
```

Destruct a Coin wrapper and keep the balance.

```bash
```

```bash
```

Take a Coin worth of value from Balance. Aborts if value > balance . value

```bash
```

```bash
```

```

Put a Coin to the Balance.

```bash

```

```bash

```

Consume the coin c and add its value to self. Aborts if c. value + self. value > U64_MAX

```bash

```

```bash

```

Split coin self to two coins, one with balance split_amount, and the remaining balance is left is self.

```bash

```

```bash

```

Split coin self into n - 1 coins with equal balances. The remainder is left in self. Return newly created coins.

```bash

```

```bash

```

Make any Coin with a zero value. Useful for placeholding bids/payments or preemptively making empty balances.

```bash

```

```bash

```

Destroy a coin with value zero

```bash

```

```bash

```

Create a new currency type T as and return the TreasuryCap for T to the caller. Can only be called with a one-time-witness type, ensuring that there's only one TreasuryCap per T.

```bash

```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. When an address is added to the deny list, it is immediately unable to interact with the currency's coin as input objects. Additionally at the start of the next epoch, they will be unable to receive the currency's coin. The allow_global_pause flag enables an additional API that will cause all addresses to be denied. Note however, that this doesn't affect per-address entries of the deny list and will not change the result of the "contains" APIs.

```bash
```

```bash
```

Given the DenyCap for a regulated currency, migrate it to the new DenyCapV2 type. All entries in the deny list will be migrated to the new format. See create_regulated_currency_v2 for details on the new v2 of the deny list.

```bash
```

```bash
```

Create a coin worth value and increase the total supply in cap accordingly.

```bash
```

```bash
```

Mint some amount of T as a Balance and increase the total supply in cap accordingly. Aborts if value + cap. total_supply >= U64_MAX

```bash
```

```bash
```

Destroy the coin c and decrease the total supply in cap accordingly.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash

```

```bash

```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash

```

```bash

```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash

```

```bash

```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash

```

```bash

```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash

```

```bash

```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash

```

```bash

```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash

```
```

```bash
```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash
```

```bash
```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash
```

```bash
```

Update name of the coin in CoinMetadata

```bash
```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

Public getter for the coin's value

```bash
```

```bash
```

Get immutable reference to the balance of a coin.

```bash
```

```bash
```

Get a mutable reference to the balance of a coin.

```bash
```

```bash
```

Wrap a balance into a Coin to make it transferable.

```bash
```

```bash
```

Destruct a Coin wrapper and keep the balance.

```bash
```

```bash
```

```

Take a Coin worth of value from Balance. Aborts if value > balance . value

```bash

```

```bash

```

Put a Coin to the Balance.

```bash

```

```bash

```

Consume the coin c and add its value to self. Aborts if c. value + self. value > U64_MAX

```bash

```

```bash

```

Split coin self to two coins, one with balance split_amount, and the remaining balance is left is self.

```bash

```

```bash

```

Split coin self into n - 1 coins with equal balances. The remainder is left in self. Return newly created coins.

```bash

```

```bash

```

Make any Coin with a zero value. Useful for placeholding bids/payments or preemptively making empty balances.

```bash

```

```bash

```

Destroy a coin with value zero

```bash

```
```

```bash
```

Create a new currency type T as and return the TreasuryCap for T to the caller. Can only be called with a one-time-witness type, ensuring that there's only one TreasuryCap per T.

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. When an address is added to the deny list, it is immediately unable to interact with the currency's coin as input objects. Additionally at the start of the next epoch, they will be unable to receive the currency's coin. The allow_global_pause flag enables an additional API that will cause all addresses to be denied. Note however, that this doesn't affect per-address entries of the deny list and will not change the result of the "contains" APIs.

```bash
```

```bash
```

Given the DenyCap for a regulated currency, migrate it to the new DenyCapV2 type. All entries in the deny list will be migrated to the new format. See create_regulated_currency_v2 for details on the new v2 of the deny list.

```bash
```

```bash
```

Create a coin worth value and increase the total supply in cap accordingly.

```bash
```

```bash
```

Mint some amount of T as a Balance and increase the total supply in cap accordingly. Aborts if value + cap. total_supply >= U64_MAX

```bash
```

```bash
```

Destroy the coin c and decrease the total supply in cap accordingly.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash
```

```bash
```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash
```

```bash
```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash
```

```bash
```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash
```

```bash
```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash
```

```bash
```

Update name of the coin in CoinMetadata

```bash
```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.
```

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

Get immutable reference to the balance of a coin.

```bash
```

```bash
```

Get a mutable reference to the balance of a coin.

```bash
```

```bash
```

Wrap a balance into a Coin to make it transferable.

```bash
```

```bash
```

Destruct a Coin wrapper and keep the balance.

```bash
```

```bash
```

```

Take a Coin worth of value from Balance. Aborts if value > balance . value

```bash

```

```bash

```

Put a Coin to the Balance.

```bash

```

```bash

```

Consume the coin c and add its value to self. Aborts if c. value + self. value > U64_MAX

```bash

```

```bash

```

Split coin self to two coins, one with balance split_amount, and the remaining balance is left is self.

```bash

```

```bash

```

Split coin self into n - 1 coins with equal balances. The remainder is left in self. Return newly created coins.

```bash

```

```bash

```

Make any Coin with a zero value. Useful for placeholding bids/payments or preemptively making empty balances.

```bash

```

```bash

```

Destroy a coin with value zero

```bash

```
```

```bash
```

Create a new currency type T as and return the TreasuryCap for T to the caller. Can only be called with a one-time-witness type, ensuring that there's only one TreasuryCap per T.

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. When an address is added to the deny list, it is immediately unable to interact with the currency's coin as input objects. Additionally at the start of the next epoch, they will be unable to receive the currency's coin. The allow_global_pause flag enables an additional API that will cause all addresses to be denied. Note however, that this doesn't affect per-address entries of the deny list and will not change the result of the "contains" APIs.

```bash
```

```bash
```

Given the DenyCap for a regulated currency, migrate it to the new DenyCapV2 type. All entries in the deny list will be migrated to the new format. See create_regulated_currency_v2 for details on the new v2 of the deny list.

```bash
```

```bash
```

Create a coin worth value and increase the total supply in cap accordingly.

```bash
```

```bash
```

Mint some amount of T as a Balance and increase the total supply in cap accordingly. Aborts if value + cap. total_supply >= U64_MAX

```bash
```

```bash
```

Destroy the coin c and decrease the total supply in cap accordingly.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash
```

```bash
```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash
```

```bash
```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash
```

```bash
```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash
```

```bash
```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash
```

```bash
```

Update name of the coin in CoinMetadata

```bash
```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.
```

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

Get a mutable reference to the balance of a coin.

```bash
```

```bash
```

Wrap a balance into a Coin to make it transferable.

```bash
```

```bash
```

Destruct a Coin wrapper and keep the balance.

```bash
```

```bash
```

Take a Coin worth of value from Balance. Aborts if value > balance . value

```bash
```

```bash
```

```

Put a Coin to the Balance.

```bash

```

```bash

```

Consume the coin c and add its value to self. Aborts if c. value + self. value > U64_MAX

```bash

```

```bash

```

Split coin self to two coins, one with balance split_amount, and the remaining balance is left is self.

```bash

```

```bash

```

Split coin self into n - 1 coins with equal balances. The remainder is left in self. Return newly created coins.

```bash

```

```bash

```

Make any Coin with a zero value. Useful for placeholding bids/payments or preemptively making empty balances.

```bash

```

```bash

```

Destroy a coin with value zero

```bash

```

```bash

```

Create a new currency type T as and return the TreasuryCap for T to the caller. Can only be called with a one-time-witness type, ensuring that there's only one TreasuryCap per T.

```bash

```

```bash

```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. When an address is added to the deny list, it is immediately unable to interact with the currency's coin as input objects. Additionally at the start of the next epoch, they will be unable to receive the currency's coin. The allow_global_pause flag enables an additional API that will cause all addresses to be denied. Note however, that this doesn't affect per-address entries of the deny list and will not change the result of the "contains" APIs.

```bash

```

```bash

```

Given the DenyCap for a regulated currency, migrate it to the new DenyCapV2 type. All entries in the deny list will be migrated to the new format. See create_regulated_currency_v2 for details on the new v2 of the deny list.

```bash

```

```bash

```

Create a coin worth value and increase the total supply in cap accordingly.

```bash

```

```bash

```

Mint some amount of T as a Balance and increase the total supply in cap accordingly. Aborts if value + cap. total_supply >= U64_MAX

```bash

```

```bash

```

Destroy the coin c and decrease the total supply in cap accordingly.

```bash

```

```bash

```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash

```

```bash

```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash

```

```bash

```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash

```

```bash

```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash

```

```bash

```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash

```

```bash

```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash

```

```bash

```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash

```
```

```bash
```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash
```

```bash
```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash
```

```bash
```

Update name of the coin in CoinMetadata

```bash
```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

Wrap a balance into a Coin to make it transferable.

```bash
```

```bash
```

Destruct a Coin wrapper and keep the balance.

```bash
```

```bash
```

Take a Coin worth of value from Balance. Aborts if value > balance . value

```bash
```

```bash
```

Put a Coin to the Balance.

```bash
```

```bash
```

Consume the coin c and add its value to self. Aborts if c. value + self. value > U64_MAX

```bash
```

```bash
```

```

Split coin self to two coins, one with balance split_amount, and the remaining balance is left is self.

```bash

```

```bash

```

Split coin self into n - 1 coins with equal balances. The remainder is left in self. Return newly created coins.

```bash

```

```bash

```

Make any Coin with a zero value. Useful for placeholding bids/payments or preemptively making empty balances.

```bash

```

```bash

```

Destroy a coin with value zero

```bash

```

```bash

```

Create a new currency type T as and return the TreasuryCap for T to the caller. Can only be called with a one-time-witness type, ensuring that there's only one TreasuryCap per T.

```bash

```

```bash

```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. When an address is added to the deny list, it is immediately unable to interact with the currency's coin as input objects. Additionally at the start of the next epoch, they will be unable to receive the currency's coin. The allow_global_pause flag enables an additional API that will cause all addresses to be denied. Note however, that this doesn't affect per-address entries of the deny list and will not change the result of the "contains" APIs.

```bash

```

```bash

```
```

Given the DenyCap for a regulated currency, migrate it to the new DenyCapV2 type. All entries in the deny list will be migrated to the new format. See create_regulated_currency_v2 for details on the new v2 of the deny list.

```bash
```

```bash
```

Create a coin worth value and increase the total supply in cap accordingly.

```bash
```

```bash
```

Mint some amount of T as a Balance and increase the total supply in cap accordingly. Aborts if value + cap. total_supply >= U64_MAX

```bash
```

```bash
```

Destroy the coin c and decrease the total supply in cap accordingly.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash
```

```bash
```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash
```

```bash
```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash
```

```bash
```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash
```

```bash
```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash
```

```bash
```

Update name of the coin in CoinMetadata

```bash
```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

Destruct a Coin wrapper and keep the balance.

```bash
```

```bash
```

Take a Coin worth of value from Balance. Aborts if value > balance . value

```bash
```

```bash
```

Put a Coin to the Balance.

```bash
```

```bash
```

Consume the coin c and add its value to self. Aborts if c. value + self. value > U64_MAX

```bash
```

```bash
```

Split coin self to two coins, one with balance split_amount, and the remaining balance is left is self.

```bash
```

```bash
```

Split coin self into n - 1 coins with equal balances. The remainder is left in self. Return newly created coins.

```bash
```

```bash
```

Make any Coin with a zero value. Useful for placeholding bids/payments or preemptively making empty balances.

```bash
```

```bash
```

```

Destroy a coin with value zero

```bash

```

```bash

```

Create a new currency type T as and return the TreasuryCap for T to the caller. Can only be called with a one-time-witness type, ensuring that there's only one TreasuryCap per T.

```bash

```

```bash

```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. When an address is added to the deny list, it is immediately unable to interact with the currency's coin as input objects. Additionally at the start of the next epoch, they will be unable to receive the currency's coin. The allow_global_pause flag enables an additional API that will cause all addresses to be denied. Note however, that this doesn't affect per-address entries of the deny list and will not change the result of the "contains" APIs.

```bash

```

```bash

```

Given the DenyCap for a regulated currency, migrate it to the new DenyCapV2 type. All entries in the deny list will be migrated to the new format. See create_regulated_currency_v2 for details on the new v2 of the deny list.

```bash

```

```bash

```

Create a coin worth value and increase the total supply in cap accordingly.

```bash

```

```bash

```

Mint some amount of T as a Balance and increase the total supply in cap accordingly. Aborts if value + cap. total_supply >= U64_MAX

```bash

```

```bash

```

Destroy the coin c and decrease the total supply in cap accordingly.

```bash

```

```bash

```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash

```

```bash

```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash

```

```bash

```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash

```

```bash

```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash

```

```bash

```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash

```

```bash

```
```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash
```

```bash
```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash
```

```bash
```

Update name of the coin in CoinMetadata

```bash
```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

Take a Coin worth of value from Balance. Aborts if value > balance . value

```bash
```

```bash
```

Put a Coin to the Balance.

```bash
```

```bash
```

Consume the coin c and add its value to self. Aborts if c. value + self. value > U64_MAX

```bash
```

```bash
```
```

Split coin self to two coins, one with balance split_amount, and the remaining balance is left is self.

```bash
```

```bash
```

Split coin self into n - 1 coins with equal balances. The remainder is left in self. Return newly created coins.

```bash
```

```bash
```

Make any Coin with a zero value. Useful for placeholding bids/payments or preemptively making empty balances.

```bash
```

```bash
```

Destroy a coin with value zero

```bash
```

```bash
```

Create a new currency type T as and return the TreasuryCap for T to the caller. Can only be called with a one-time-witness type, ensuring that there's only one TreasuryCap per T.

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. When an address is added to the deny list, it is immediately unable to interact with the currency's coin as input objects. Additionally at the start of the next epoch, they will be unable to receive the currency's coin. The allow_global_pause flag enables an additional API that will cause all addresses to be denied. Note however, that this doesn't affect per-address entries of the deny list and will not change the result of the "contains" APIs.

```bash
```

```bash
```

Given the DenyCap for a regulated currency, migrate it to the new DenyCapV2 type. All entries in the deny list will be migrated to the new format. See create_regulated_currency_v2 for details on the new v2 of the deny list.

```bash
```

```bash
```

Create a coin worth value and increase the total supply in cap accordingly.

```bash
```

```bash
```

Mint some amount of T as a Balance and increase the total supply in cap accordingly. Aborts if value + cap. total_supply >= U64_MAX

```bash
```

```bash
```

Destroy the coin c and decrease the total supply in cap accordingly.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash
```

```bash
```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash
```

```bash
```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash
```

```bash
```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash
```

```bash
```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash
```

```bash
```

Update name of the coin in CoinMetadata

```bash
```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

Put a Coin to the Balance.

```bash
```

```

```bash
```

Consume the coin c and add its value to self. Aborts if c. value + self. value > U64_MAX

```bash
```

```bash
```

Split coin self to two coins, one with balance split_amount, and the remaining balance is left is self.

```bash
```

```bash
```

Split coin self into n - 1 coins with equal balances. The remainder is left in self. Return newly created coins.

```bash
```

```bash
```

Make any Coin with a zero value. Useful for placeholding bids/payments or preemptively making empty balances.

```bash
```

```bash
```

Destroy a coin with value zero

```bash
```

```bash
```

Create a new currency type T as and return the TreasuryCap for T to the caller. Can only be called with a one-time-witness type, ensuring that there's only one TreasuryCap per T.

```bash
```

```bash
```
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. When an address is added to the deny list, it is immediately unable to interact with the currency's coin as input objects. Additionally at the start of the next epoch, they will be unable to receive the currency's coin. The allow_global_pause flag enables an additional API that will cause all addresses to be denied. Note however, that this doesn't affect per-address entries of the deny list and will not change the result of the "contains" APIs.

```bash
```

```bash
```

Given the DenyCap for a regulated currency, migrate it to the new DenyCapV2 type. All entries in the deny list will be migrated to the new format. See create_regulated_currency_v2 for details on the new v2 of the deny list.

```bash
```

```bash
```

Create a coin worth value and increase the total supply in cap accordingly.

```bash
```

```bash
```

Mint some amount of T as a Balance and increase the total supply in cap accordingly. Aborts if value + cap. total_supply >= U64_MAX

```bash
```

```bash
```

Destroy the coin c and decrease the total supply in cap accordingly.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash

```

```bash

```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash

```

```bash

```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash

```

```bash

```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash

```

```bash

```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash

```

```bash

```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash

```

```bash

```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash
```

```bash
```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash
```

```bash
```

Update name of the coin in CoinMetadata

```bash
```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```
```

```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash

```

```bash

```

Function

Consume the coin c and add its value to self. Aborts if c. value + self. value > U64_MAX

```bash

```

```bash

```

Split coin self to two coins, one with balance split_amount, and the remaining balance is left is self.

```bash

```

```bash

```

Split coin self into n - 1 coins with equal balances. The remainder is left in self. Return newly created coins.

```bash

```

```bash

```

Make any Coin with a zero value. Useful for placeholding bids/payments or preemptively making empty balances.

```bash

```

```bash

```

Destroy a coin with value zero

```bash

```

```bash

```

Create a new currency type T as and return the TreasuryCap for T to the caller. Can only be called with a one-time-witness type, ensuring that there's only one TreasuryCap per T.
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. When an address is added to the deny list, it is immediately unable to interact with the currency's coin as input objects. Additionally at the start of the next epoch, they will be unable to receive the currency's coin. The allow_global_pause flag enables an additional API that will cause all addresses to be denied. Note however, that this doesn't affect per-address entries of the deny list and will not change the result of the "contains" APIs.

```bash
```

```bash
```

Given the DenyCap for a regulated currency, migrate it to the new DenyCapV2 type. All entries in the deny list will be migrated to the new format. See create_regulated_currency_v2 for details on the new v2 of the deny list.

```bash
```

```bash
```

Create a coin worth value and increase the total supply in cap accordingly.

```bash
```

```bash
```

Mint some amount of T as a Balance and increase the total supply in cap accordingly. Aborts if value + cap. total_supply >= U64_MAX

```bash
```

```bash
```

Destroy the coin c and decrease the total supply in cap accordingly.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash
```

```bash
```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash
```

```bash
```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash
```

```bash
```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash
```

```

```bash
```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash
```

```bash
```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash
```

```bash
```

Update name of the coin in CoinMetadata

```bash
```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

Split coin self to two coins, one with balance split_amount, and the remaining balance is left is self.

```bash
```

```bash
```

Split coin self into n - 1 coins with equal balances. The remainder is left in self. Return newly created coins.

```bash
```

```bash
```

Make any Coin with a zero value. Useful for placeholding bids/payments or preemptively making empty balances.

```bash
```

```bash
```

Destroy a coin with value zero

```bash
```

```bash
```

Create a new currency type T as and return the TreasuryCap for T to the caller. Can only be called with a one-time-witness type, ensuring that there's only one TreasuryCap per T.

```bash
```

```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. When an address is added to the deny list, it is immediately unable to interact with the currency's coin as input objects. Additionally at the start of the next epoch, they will be unable to receive the currency's coin. The allow_global_pause flag enables an additional API that will cause all addresses to be denied. Note however, that this doesn't affect per-address entries of the deny list and will not change the result of the "contains" APIs.

```bash
```

```bash
```

Given the DenyCap for a regulated currency, migrate it to the new DenyCapV2 type. All entries in the deny list will be migrated to the new format. See create_regulated_currency_v2 for details on the new v2 of the deny list.

```bash
```

```bash
```

Create a coin worth value and increase the total supply in cap accordingly.

```bash
```

```bash
```

Mint some amount of T as a Balance and increase the total supply in cap accordingly. Aborts if value + cap. total_supply >= U64_MAX

```bash
```

```bash
```

Destroy the coin c and decrease the total supply in cap accordingly.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash

```

```bash

```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash

```

```bash

```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash

```

```bash

```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash

```

```bash

```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash

```

```bash

```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash

```

```bash

```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash

```
```

```bash
```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash
```

```bash
```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash
```

```bash
```

Update name of the coin in CoinMetadata

```bash
```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

Split coin self into n - 1 coins with equal balances. The remainder is left in self. Return newly created coins.

```bash
```

```bash
```

Make any Coin with a zero value. Useful for placeholding bids/payments or preemptively making empty balances.

```bash
```

```bash
```

Destroy a coin with value zero

```bash
```

```bash
```

Create a new currency type T as and return the TreasuryCap for T to the caller. Can only be called with a one-time-witness type, ensuring that there's only one TreasuryCap per T.

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. When an address is added to the deny list, it is immediately unable to interact with the currency's coin as input objects. Additionally at the start of the next epoch, they will be unable to receive the currency's coin. The allow_global_pause flag enables an additional API that will cause all addresses to be denied. Note however, that this doesn't affect per-address entries of the deny list and will not change the result of the "contains" APIs.

```bash
```

```bash
```

Given the DenyCap for a regulated currency, migrate it to the new DenyCapV2 type. All entries in the deny list will be migrated to the new format. See create_regulated_currency_v2 for details on the new v2 of the deny list.

```bash
```

```bash
```

Create a coin worth value and increase the total supply in cap accordingly.

```bash
```

```bash
```

Mint some amount of T as a Balance and increase the total supply in cap accordingly. Aborts if value + cap. total_supply >= U64_MAX

```bash
```

```bash
```

Destroy the coin c and decrease the total supply in cap accordingly.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash
```

```bash
```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash
```

```bash
```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash
```

```bash
```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash
```

```bash
```

```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash

```

```bash

```

Update name of the coin in CoinMetadata

```bash

```

```bash

```

Update the symbol of the coin in CoinMetadata

```bash

```

```bash

```

Update the description of the coin in CoinMetadata

```bash

```

```bash

```

Update the url of the coin in CoinMetadata

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```

```bash

```

Function

Make any Coin with a zero value. Useful for placeholding bids/payments or preemptively making empty balances.

```bash

```

```bash

```

Destroy a coin with value zero

```bash

```

```bash

```

Create a new currency type T as and return the TreasuryCap for T to the caller. Can only be called with a one-time-witness type, ensuring that there's only one TreasuryCap per T.

```bash

```

```bash

```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. When an address is added to the deny list, it is immediately unable to interact with the currency's coin as input objects. Additionally at the start of the next epoch, they will be unable to receive the currency's coin. The allow_global_pause flag enables an additional API that will cause all addresses to be denied. Note however, that this doesn't affect per-address entries of the deny list and will not change the result of the "contains" APIs.

```bash

```

```bash

```

Given the DenyCap for a regulated currency, migrate it to the new DenyCapV2 type. All entries in the deny list will be migrated to the new format. See create_regulated_currency_v2 for details on the new v2 of the deny list.

```bash

```

```bash

```

Create a coin worth value and increase the total supply in cap accordingly.
```

```bash
```

```bash
```

Mint some amount of T as a Balance and increase the total supply in cap accordingly. Aborts if value + cap. total_supply >= U64_MAX

```bash
```

```bash
```

Destroy the coin c and decrease the total supply in cap accordingly.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash
```

```bash
```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash
```

```bash
```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```

```bash

```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash

```

```bash

```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash

```

```bash

```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash

```

```bash

```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash

```

```bash

```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash

```

```bash

```

Update name of the coin in CoinMetadata

```bash

```

```bash

```
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

Destroy a coin with value zero

```bash
```

```bash
```

Create a new currency type T as and return the TreasuryCap for T to the caller. Can only be called with a one-time-witness type, ensuring that there's only one TreasuryCap per T.
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. When an address is added to the deny list, it is immediately unable to interact with the currency's coin as input objects. Additionally at the start of the next epoch, they will be unable to receive the currency's coin. The allow_global_pause flag enables an additional API that will cause all addresses to be denied. Note however, that this doesn't affect per-address entries of the deny list and will not change the result of the "contains" APIs.

```bash
```

```bash
```

Given the DenyCap for a regulated currency, migrate it to the new DenyCapV2 type. All entries in the deny list will be migrated to the new format. See create_regulated_currency_v2 for details on the new v2 of the deny list.

```bash
```

```bash
```

Create a coin worth value and increase the total supply in cap accordingly.

```bash
```

```bash
```

Mint some amount of T as a Balance and increase the total supply in cap accordingly. Aborts if value + cap. total_supply >= U64_MAX

```bash
```

```bash
```

Destroy the coin c and decrease the total supply in cap accordingly.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash
```

```bash
```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash
```

```bash
```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash
```

```bash
```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash
```

```

```bash
```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash
```

```bash
```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash
```

```bash
```

Update name of the coin in CoinMetadata

```bash
```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

Create a new currency type T as and return the TreasuryCap for T to the caller. Can only be called with a one-time-witness type, ensuring that there's only one TreasuryCap per T.

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. When an address is added to the deny list, it is immediately unable to interact with the currency's coin as input objects. Additionally at the start of the next epoch, they will be unable to receive the currency's coin. The allow_global_pause flag enables an additional API that will cause all addresses to be denied. Note however, that this doesn't affect per-address entries of the deny list and will not change the result of the "contains" APIs.

```bash
```

```bash
```

Given the DenyCap for a regulated currency, migrate it to the new DenyCapV2 type. All entries in the deny list will be migrated to the new format. See create_regulated_currency_v2 for details on the new v2 of the deny list.

```bash
```

```bash
```

Create a coin worth value and increase the total supply in cap accordingly.

```bash
```

```bash
```

```

Mint some amount of T as a Balance and increase the total supply in cap accordingly. Aborts if value + cap. total_supply >= U64_MAX

```bash

```

```bash

```

Destroy the coin c and decrease the total supply in cap accordingly.

```bash

```

```bash

```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash

```

```bash

```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash

```

```bash

```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash

```

```bash

```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash

```

```bash

```
```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash
```

```bash
```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash
```

```bash
```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash
```

```bash
```

Update name of the coin in CoinMetadata

```bash
```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash

```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash

```

```bash

```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash

```

```bash

```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash

```

```bash

```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash

```

```bash

```

Function

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. When an address is added to the deny list, it is immediately unable to interact with the currency's coin as input objects. Additionally at the start of the next epoch, they will be unable to receive the currency's coin. The allow_global_pause flag enables an additional API that will cause all addresses to be denied. Note however, that this doesn't affect per-address entries of the deny list and will not change the result of the "contains" APIs.

```bash

```

```bash

```

Given the DenyCap for a regulated currency, migrate it to the new DenyCapV2 type. All entries in the deny list will be migrated to the new format. See create_regulated_currency_v2 for details on the new v2 of the deny list.

```bash
```

```

```bash
```

Create a coin worth value and increase the total supply in cap accordingly.

```bash
```

```bash
```

Mint some amount of T as a Balance and increase the total supply in cap accordingly. Aborts if value + cap. total_supply >= U64_MAX

```bash
```

```bash
```

Destroy the coin c and decrease the total supply in cap accordingly.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash
```

```bash
```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash
```

```bash
```
```

```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash

```

```bash

```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash

```

```bash

```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash

```

```bash

```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash

```

```bash

```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash

```

```bash

```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash

```

```bash

```

Update name of the coin in CoinMetadata
```

```bash
```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

Given the DenyCap for a regulated currency, migrate it to the new DenyCapV2 type. All entries in the deny list will be migrated to the new format. See create_regulated_currency_v2 for details on the new v2 of the deny list.

```bash
```
```

```

```bash
```

Create a coin worth value and increase the total supply in cap accordingly.

```bash
```

```bash
```

Mint some amount of T as a Balance and increase the total supply in cap accordingly. Aborts if value + cap. total_supply >= U64_MAX

```bash
```

```bash
```

Destroy the coin c and decrease the total supply in cap accordingly.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash
```

```bash
```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash
```

```bash
```
```

```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash

```

```bash

```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash

```

```bash

```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash

```

```bash

```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash

```

```bash

```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash

```

```bash

```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash

```

```bash

```

Update name of the coin in CoinMetadata
```

```bash
```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

Create a coin worth value and increase the total supply in cap accordingly.

```bash
```
```

```bash
```

Mint some amount of T as a Balance and increase the total supply in cap accordingly. Aborts if value + cap. total_supply >= U64_MAX

```bash
```

```bash
```

Destroy the coin c and decrease the total supply in cap accordingly.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash
```

```bash
```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash
```

```bash
```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash

```

```bash

```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash

```

```bash

```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash

```

```bash

```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash

```

```bash

```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash

```

```bash

```

Update name of the coin in CoinMetadata

```bash

```

```bash

```

Update the symbol of the coin in CoinMetadata

```bash

```

```bash

```

Update the description of the coin in CoinMetadata

```bash

```

```bash

```

Update the url of the coin in CoinMetadata

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash

```

```bash

```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash

```

```bash

```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash

```

```bash

```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash

```

```bash

```

Function

Mint some amount of T as a Balance and increase the total supply in cap accordingly. Aborts if value + cap. total_supply >= U64_MAX

```bash

```

```bash

```

Destroy the coin c and decrease the total supply in cap accordingly.

```bash

```
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash
```

```bash
```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash
```

```bash
```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash
```

```bash
```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash
```

```bash
```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash
```

```bash
```

Update name of the coin in CoinMetadata

```bash
```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.
```

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

Destroy the coin c and decrease the total supply in cap accordingly.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction. Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash
```

```bash
```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash
```

```

```bash

```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash

```

```bash

```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash

```

```bash

```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash

```

```bash

```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash

```

```bash

```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash

```

```bash

```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash

```

```bash

```

Update name of the coin in CoinMetadata

```bash

```

```bash

```

Update the symbol of the coin in CoinMetadata

```bash

```

```bash

```

Update the description of the coin in CoinMetadata

```bash

```

```bash

```

Update the url of the coin in CoinMetadata

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.
```

Additionally at the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash
```

```bash
```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash
```

```bash
```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash
```

```bash
```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash
```

```bash
```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash
```

```bash
```

Update name of the coin in CoinMetadata

```bash
```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash

```

```bash

```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash

```

```bash

```

Function

Removes an address from the deny list. Similar to deny_list_v2_add , the effect for input objects will be immediate, but the effect for receiving objects will be delayed until the next epoch.

```bash

```

```bash

```

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash

```

```bash

```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash

```

```bash

```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash

```

```bash

```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash

```

```bash

```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash

```

```bash

```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash

```

```bash

```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash

```

```bash

```

Update name of the coin in CoinMetadata

```bash

```

```bash

```

Update the symbol of the coin in CoinMetadata

```bash

```

```bash

```

Update the description of the coin in CoinMetadata

```bash

```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.
```

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

Check if the deny list contains the given address for the current epoch. Denied addresses in the current epoch will be unable to receive objects of this coin type.

```bash
```

```bash
```

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash
```

```bash
```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash
```

```bash
```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash
```

```bash
```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash
```

```bash
```

Update name of the coin in CoinMetadata

```bash
```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

```

Update the description of the coin in CoinMetadata

```bash

```

```bash

```

Update the url of the coin in CoinMetadata

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash

```

```bash

```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash

```

```bash

```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash

```

```bash

```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash

```

```bash

```

Function

Check if the deny list contains the given address for the next epoch. Denied addresses in the next epoch will immediately be unable to use objects of this coin type as inputs. At the start of the next epoch, the address will be unable to receive objects of this coin type.

```bash

```

```bash

```

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash

```

```bash

```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash

```

```bash

```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash

```

```bash

```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash

```

```bash

```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash

```

```bash

```

Update name of the coin in CoinMetadata

```bash

```

```bash

```

Update the symbol of the coin in CoinMetadata

```bash

```

```bash

```

Update the description of the coin in CoinMetadata

```bash

```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.
```

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

Enable the global pause for the given coin type. This will immediately prevent all addresses from using objects of this coin type as inputs. At the start of the next epoch, all addresses will be unable to receive objects of this coin type.

```bash
```

```bash
```

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash
```

```

```bash
```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash
```

```bash
```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash
```

```bash
```

Update name of the coin in CoinMetadata

```bash
```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

Disable the global pause for the given coin type. This will immediately allow all addresses to resume using objects of this coin type as inputs. However, receiving objects of this coin type will still be paused until the start of the next epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the current epoch.

```bash
```

```bash
```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash
```

```bash
```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash
```

```bash
```

Update name of the coin in CoinMetadata

```bash
```

```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

Check if the global pause is enabled for the given coin type in the current epoch.

```bash
```

```bash
```

```

Check if the global pause is enabled for the given coin type in the next epoch.

```bash

```

```bash

```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash

```

```bash

```

Update name of the coin in CoinMetadata

```bash

```

```bash

```

Update the symbol of the coin in CoinMetadata

```bash

```

```bash

```

Update the description of the coin in CoinMetadata

```bash

```

```bash

```

Update the url of the coin in CoinMetadata

```bash

```

```bash

```

```bash

```

```bash

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```
```

```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

Check if the global pause is enabled for the given coin type in the next epoch.

```bash
```

```bash
```

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash
```

```bash
```

Update name of the coin in CoinMetadata

```bash
```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash

```

```bash

```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash

```

```bash

```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash

```

```bash

```

Function

Mint amount of Coin and send it to recipient. Invokes mint ().

```bash

```

```bash

```

Update name of the coin in CoinMetadata

```bash

```

```bash

```

Update the symbol of the coin in CoinMetadata

```bash

```

```bash

```

Update the description of the coin in CoinMetadata

```bash

```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.
```

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

Update name of the coin in CoinMetadata

```bash
```

```bash
```

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

Update the symbol of the coin in CoinMetadata

```bash
```

```bash
```

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

Update the description of the coin in CoinMetadata

```bash
```

```bash
```

Update the url of the coin in CoinMetadata

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

Update the url of the coin in CoinMetadata
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

```

## Function

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

```bash
```

```bash
```

```bash
```

```bash
```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash
```

```bash
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

```

## Function

```bash

```

```bash

```

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash

```

```bash

```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash

```

```bash

```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash

```

```bash

```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash

```

```bash

```

Function

This creates a new currency, via create_currency , but with an extra capability that allows for specific addresses to have their coins frozen. Those addresses cannot interact with the coin as input objects.

```bash

```

```bash

```
```

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

Adds the given address to the deny list, preventing it from interacting with the specified coin type as an input to a transaction.

```bash
```

```bash
```

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

Removes an address from the deny list. Aborts with ENotFrozen if the address is not already in the list.

```bash
```

```bash
```

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```

Function

Returns true iff the given address is denied for the given coin type. It will return false if given a non-coin type.

```bash
```

```bash
```