Module sui::deny_list

Defines the [DenyList](#) type. The [DenyList](#) shared object is used to restrict access to instances of certain core types from being used as inputs by specified addresses in the deny list.

A shared object that stores the addresses that are blocked for a given core type.

The capability used to write to the deny list config. Ensures that the Configs for the DenyList are modified only by this module.

The dynamic object field key used to store the Config for a given type, essentially a (per_type_index, per_type_key) pair.

The setting key used to store the deny list for a given address in the Config.

The setting key used to store the global pause setting in the Config.

The event emitted when a new Config is created for a given type. This can be useful for tracking the ID of a type's Config object.

Stores the addresses that are denied for a given core type.

Trying to create a deny list object when not called by the system address.

The specified address to be removed is not already in the deny list.

The specified address cannot be added to the deny list.

The index into the deny list vector for the [sui::coin::Coin](#) type.

These addresses are reserved and cannot be added to the deny list. The addresses listed are well known package and object addresses. So it would be meaningless to add them to the deny list.

Adds the given address to the deny list of the specified type, preventing it from interacting with instances of that type as an input to a transaction. For coins, the type specified is the type of the coin, not the coin type itself. For example, "00...0123::my_coin::MY_COIN" would be the type, not "00...02::coin::Coin".

Removes a previously denied address from the list. Aborts with [ENotDenied](#) if the address is not on the list.

Returns true iff the given address is denied for the given type.

Creation of the deny list object is restricted to the system address via a system transaction.

Struct

A shared object that stores the addresses that are blocked for a given core type.

```bash

```

The capability used to write to the deny list config. Ensures that the Configs for the DenyList are modified only by this module.

```bash

```

The dynamic object field key used to store the Config for a given type, essentially a (per_type_index, per_type_key) pair.

```bash

```

The setting key used to store the deny list for a given address in the Config.

```bash

```

The setting key used to store the global pause setting in the Config.

```bash
```

The event emitted when a new Config is created for a given type. This can be useful for tracking the ID of a type's Config object.

```bash
```

Stores the addresses that are denied for a given core type.

```bash
```

Trying to create a deny list object when not called by the system address.

```bash
```

The specified address to be removed is not already in the deny list.

```bash
```

The specified address cannot be added to the deny list.

```bash
```

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

```bash
```

These addresses are reserved and cannot be added to the deny list. The addresses listed are well known package and object addresses. So it would be meaningless to add them to the deny list.

```bash
```

```
bash (
```

```
bash (
```

```
bash (
```

```
bash (
```

```
bash (
```

```
bash (
```

```
bash (
```

```
bash (
```

```
bash (
```

```
bash (
```

```
bash (
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

Adds the given address to the deny list of the specified type, preventing it from interacting with instances of that type as an input to a transaction. For coins, the type specified is the type of the coin, not the coin type itself. For example, "00...0123::my_coin::MY_COIN" would be the type, not "00...02::coin::Coin".

```bash
(
```

```bash
(
```

```bash

```

```bash

```

Removes a previously denied address from the list. Aborts with [ENotDenied](#) if the address is not on the list.

```bash
(
```

```bash
(
```

```bash
```

```bash
```

Returns true iff the given address is denied for the given type.

```bash
(
```

```bash
(
```

```bash
```

```bash
```

Creation of the deny list object is restricted to the system address via a system transaction.

```bash
```

```bash
```

```bash
```

```bash
```

Struct

The capability used to write to the deny list config. Ensures that the Configs for the DenyList are modified only by this module.

```bash
```

The dynamic object field key used to store the Config for a given type, essentially a (per_type_index, per_type_key) pair.

```bash
```

The setting key used to store the deny list for a given address in the Config.

```bash
```

The setting key used to store the global pause setting in the Config.

```bash
```

The event emitted when a new Config is created for a given type. This can be useful for tracking the ID of a type's Config object.

```bash
```

Stores the addresses that are denied for a given core type.

```bash
```

Trying to create a deny list object when not called by the system address.

```bash
```

The specified address to be removed is not already in the deny list.

```bash
```

The specified address cannot be added to the deny list.

```bash
```

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

```bash
```

These addresses are reserved and cannot be added to the deny list. The addresses listed are well known package and object addresses. So it would be meaningless to add them to the deny list.

```bash
```

```
bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

Adds the given address to the deny list of the specified type, preventing it from interacting with instances of that type as an input to a transaction. For coins, the type specified is the type of the coin, not the coin type itself. For example, "00...0123::my_coin::MY_COIN" would be the type, not "00...02::coin::Coin".

```bash
(
```

```bash
(
```

```bash

```

```bash

```

Removes a previously denied address from the list. Aborts with [ENotDenied](#) if the address is not on the list.

```bash
bash (
```

```bash
bash (
```

```bash
```

```bash
```

Returns true iff the given address is denied for the given type.

```bash
bash (
```

```bash
bash (
```

```bash
```

```bash
```

Creation of the deny list object is restricted to the system address via a system transaction.

```bash
```

```bash
```

```bash
```

```bash
```

Struct

The dynamic object field key used to store the Config for a given type, essentially a (per_type_index, per_type_key) pair.

```bash
```

The setting key used to store the deny list for a given address in the Config.

```bash
```

The setting key used to store the global pause setting in the Config.

```bash
```

The event emitted when a new Config is created for a given type. This can be useful for tracking the ID of a type's Config object.

```bash
```

```

```

Stores the addresses that are denied for a given core type.

```bash

```

Trying to create a deny list object when not called by the system address.

```bash

```

The specified address to be removed is not already in the deny list.

```bash

```

The specified address cannot be added to the deny list.

```bash

```

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

```bash

```

These addresses are reserved and cannot be added to the deny list. The addresses listed are well known package and object addresses. So it would be meaningless to add them to the deny list.

```bash

```

```
bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (
```

```bash
(
```

```bash
(
```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

Adds the given address to the deny list of the specified type, preventing it from interacting with instances of that type as an input to a transaction. For coins, the type specified is the type of the coin, not the coin type itself. For example, "00...0123::my_coin::MY_COIN" would be the type, not "00...02::coin::Coin".

```bash
(
```

```bash
(
```

```bash

```

```bash

```

Removes a previously denied address from the list. Aborts with [ENotDenied](#) if the address is not on the list.

```bash
(
```

```bash
(
```

```bash

```

```bash
```

Returns true iff the given address is denied for the given type.

```
bash (
```

```
bash (
```

```bash
```

```bash
```

Creation of the deny list object is restricted to the system address via a system transaction.

```bash
```

```bash
```

```bash
```

```bash
```

Struct

The setting key used to store the deny list for a given address in the Config.

```bash
```

The setting key used to store the global pause setting in the Config.

```bash
```

The event emitted when a new Config is created for a given type. This can be useful for tracking the ID of a type's Config object.

```bash
```

Stores the addresses that are denied for a given core type.

```bash
```

Trying to create a deny list object when not called by the system address.

```bash
```

```
```

The specified address to be removed is not already in the deny list.

```bash
```

The specified address cannot be added to the deny list.

```bash
```

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

```bash
```

These addresses are reserved and cannot be added to the deny list. The addresses listed are well known package and object addresses. So it would be meaningless to add them to the deny list.

```bash
```

```
bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Adds the given address to the deny list of the specified type, preventing it from interacting with instances of that type as an input to a transaction. For coins, the type specified is the type of the coin, not the coin type itself. For example, "00...0123::my_coin::MY_COIN" would be the type, not "00...02::coin::Coin".

```
bash (
```

```
bash (
```

```bash
```

```bash
```

Removes a previously denied address from the list. Aborts with ENotDenied if the address is not on the list.

```
bash (
```

```
bash (
```

```bash
```

```bash
```

Returns true iff the given address is denied for the given type.

```
bash (
```

```
bash (
```

```bash
```

```
```

```bash
```

Creation of the deny list object is restricted to the system address via a system transaction.

```bash
```

```bash
```

```bash
```

```bash
```

Struct

The setting key used to store the global pause setting in the Config.

```bash
```

The event emitted when a new Config is created for a given type. This can be useful for tracking the ID of a type's Config object.

```bash
```

Stores the addresses that are denied for a given core type.

```bash
```

Trying to create a deny list object when not called by the system address.

```bash
```

The specified address to be removed is not already in the deny list.

```bash
```

The specified address cannot be added to the deny list.

```bash
```

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

```bash
```

```

These addresses are reserved and cannot be added to the deny list. The addresses listed are well known package and object addresses. So it would be meaningless to add them to the deny list.

```bash
```

```
bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```
```

```bash
```

```bash
```

```bash
```

Adds the given address to the deny list of the specified type, preventing it from interacting with instances of that type as an input to a transaction. For coins, the type specified is the type of the coin, not the coin type itself. For example, "00...0123::my_coin::MY_COIN" would be the type, not "00...02::coin::Coin".

```
bash (
```

```
bash (
```

```bash
```

```bash
```

Removes a previously denied address from the list. Aborts with [ENotDenied](#) if the address is not on the list.

```
bash (
```

```
bash (
```

```bash
```

```bash
```

Returns true iff the given address is denied for the given type.

```
bash (
```

```
bash (
```

```bash
```

```bash
```

Creation of the deny list object is restricted to the system address via a system transaction.

```bash
```

```bash
```

```bash
```

```
```

```bash
```

Struct

The event emitted when a new Config is created for a given type. This can be useful for tracking the ID of a type's Config object.

```bash
```

Stores the addresses that are denied for a given core type.

```bash
```

Trying to create a deny list object when not called by the system address.

```bash
```

The specified address to be removed is not already in the deny list.

```bash
```

The specified address cannot be added to the deny list.

```bash
```

The index into the deny list vector for the [sui::coin::Coin](#) type.

```bash
```

These addresses are reserved and cannot be added to the deny list. The addresses listed are well known package and object addresses. So it would be meaningless to add them to the deny list.

```bash
```

```
bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Adds the given address to the deny list of the specified type, preventing it from interacting with instances of that type as an input to a transaction. For coins, the type specified is the type of the coin, not the coin type itself. For example, "00...0123::my_coin::MY_COIN" would be the type, not "00...02::coin::Coin".

```bash
(
```

```bash
(
```

```bash
```

```bash
```

```bash

```

Removes a previously denied address from the list. Aborts with ENotDenied if the address is not on the list.

```
bash (
```

```
bash (
```

```bash

```

```bash

```

Returns true iff the given address is denied for the given type.

```
bash (
```

```
bash (
```

```bash

```

```bash

```

Creation of the deny list object is restricted to the system address via a system transaction.

```bash

```

```bash

```

```bash

```

```bash

```

Struct

Stores the addresses that are denied for a given core type.

```bash

```

Trying to create a deny list object when not called by the system address.

```bash

```

The specified address to be removed is not already in the deny list.

```bash

```

```
```

The specified address cannot be added to the deny list.

```bash
```

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

```bash
```

These addresses are reserved and cannot be added to the deny list. The addresses listed are well known package and object addresses. So it would be meaningless to add them to the deny list.

```bash
```

```
bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (
```

```bash
```

```bash
```

```bash
```

```bash
```

```
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Adds the given address to the deny list of the specified type, preventing it from interacting with instances of that type as an input to a transaction. For coins, the type specified is the type of the coin, not the coin type itself. For example, "00...0123::my_coin::MY_COIN" would be the type, not "00...02::coin::Coin".

```
bash (
```

```
bash (
```

```bash
```

```bash
```

Removes a previously denied address from the list. Aborts with [ENotDenied](#) if the address is not on the list.

```
bash (
```

```
bash (
```

```bash
```

```bash
```

Returns true iff the given address is denied for the given type.

```
bash (
```

```
bash (
```

```bash
```

```bash
```

Creation of the deny list object is restricted to the system address via a system transaction.

```bash

```

```bash

```

```bash

```

```bash

```

Constants

Trying to create a deny list object when not called by the system address.

```bash

```

The specified address to be removed is not already in the deny list.

```bash

```

The specified address cannot be added to the deny list.

```bash

```

The index into the deny list vector for the [sui::coin::Coin](#) type.

```bash

```

These addresses are reserved and cannot be added to the deny list. The addresses listed are well known package and object addresses. So it would be meaningless to add them to the deny list.

```bash

```

```
bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (
```

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

Adds the given address to the deny list of the specified type, preventing it from interacting with instances of that type as an input to a transaction. For coins, the type specified is the type of the coin, not the coin type itself. For example, "00...0123::my_coin::MY_COIN" would be the type, not "00...02::coin::Coin".

bash (

bash (

```bash

```

```bash
```

Removes a previously denied address from the list. Aborts with [ENotDenied](#) if the address is not on the list.

```bash
(
```

```bash
(
```

```bash
```

```bash
```

```bash
```

Returns true iff the given address is denied for the given type.

```bash
(
```

```bash
(
```

```bash
```

```bash
```

```bash
```

Creation of the deny list object is restricted to the system address via a system transaction.

```bash
```

```bash
```

```bash
```

```bash
```

Function

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Adds the given address to the deny list of the specified type, preventing it from interacting with instances of that type as an input to a transaction. For coins, the type specified is the type of the coin, not the coin type itself. For example, "00...0123::my_coin::MY_COIN" would be the type, not "00...02::coin::Coin".

```bash
(
```

```bash
(
```

```bash
```

```bash
```

```bash
```

Removes a previously denied address from the list. Aborts with ENotDenied if the address is not on the list.

bash (

bash (

```bash
```

```bash
```

Returns true iff the given address is denied for the given type.

bash (

bash (

```bash
```

```bash
```

Creation of the deny list object is restricted to the system address via a system transaction.

```bash
```

```bash
```

```bash
```

```bash
```

Function

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

bash (

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

Adds the given address to the deny list of the specified type, preventing it from interacting with instances of that type as an input to a transaction. For coins, the type specified is the type of the coin, not the coin type itself. For example, "00...0123::my_coin::MY_COIN" would be the type, not "00...02::coin::Coin".

```bash
(
```

```bash
(
```

```bash

```

```bash

```

Removes a previously denied address from the list. Aborts with [ENotDenied](#) if the address is not on the list.

```bash
(
```

```bash
(
```

```bash

```

```bash

```

Returns true iff the given address is denied for the given type.

```bash
(
```

```bash
(
```

```bash

```

```bash

```

Creation of the deny list object is restricted to the system address via a system transaction.

```bash

```

```bash

```

```bash

```

```bash

```

Function

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Adds the given address to the deny list of the specified type, preventing it from interacting with instances of that type as an input to a transaction. For coins, the type specified is the type of the coin, not the coin type itself. For example, "00...0123::my_coin::MY_COIN" would be the type, not "00...02::coin::Coin".

```bash
(
```

```bash
(
```

```bash
```

```bash
```

Removes a previously denied address from the list. Aborts with [ENotDenied](#) if the address is not on the list.

```bash
(
```

```bash
(
```

```bash
```

```
```

```bash
```

Returns true iff the given address is denied for the given type.

```
bash (
```

```
bash (
```

```bash
```

```bash
```

Creation of the deny list object is restricted to the system address via a system transaction.

```bash
```

```bash
```

```bash
```

```bash
```

Function

```
bash (
```

```
bash (
```

```
bash (
```

```
bash (
```

```
bash (
```

```
bash (
```

```
bash (
```

```
bash (
```

```
bash (
```

```
bash (
```

```
bash (
```

```
bash (
```

```bash
```

```bash
```

```
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Adds the given address to the deny list of the specified type, preventing it from interacting with instances of that type as an input to a transaction. For coins, the type specified is the type of the coin, not the coin type itself. For example, "00...0123::my_coin::MY_COIN" would be the type, not "00...02::coin::Coin".

```bash (
```

```bash (
```

```bash
```

```bash
```

Removes a previously denied address from the list. Aborts with [ENotDenied](#) if the address is not on the list.

```bash (
```

```bash (
```

```bash
```

```bash
```

Returns true iff the given address is denied for the given type.

```bash (
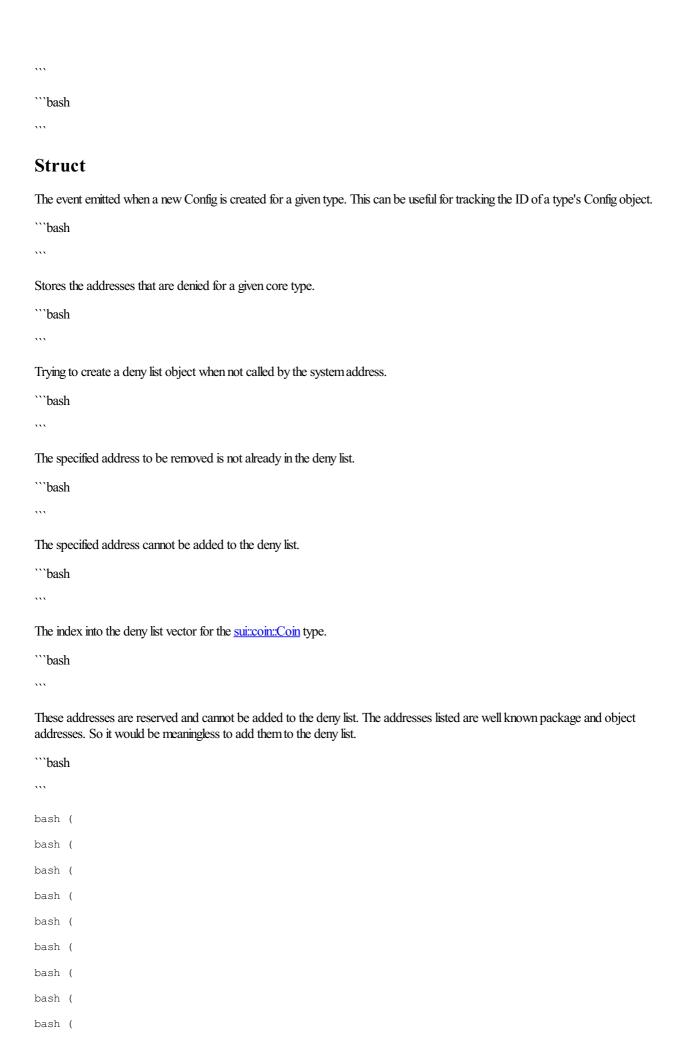```

```bash
(
```

```bash

```

```bash

```

Creation of the deny list object is restricted to the system address via a system transaction.

```bash

```

```bash

```

```bash

```

```bash

```

Function

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Adds the given address to the deny list of the specified type, preventing it from interacting with instances of that type as an input to a transaction. For coins, the type specified is the type of the coin, not the coin type itself. For example, "00...0123::my_coin::MY_COIN" would be the type, not "00...02::coin::Coin".

```
bash (
```

```
bash (
```

```bash
```

```bash
```

Removes a previously denied address from the list. Aborts with [ENotDenied](#) if the address is not on the list.

```
bash (
```

```
bash (
```

```bash
```

```bash
```

Returns true iff the given address is denied for the given type.

```
bash (
```

```
bash (
```

```bash
```

```bash
```

Creation of the deny list object is restricted to the system address via a system transaction.

```bash
```

```bash
```

```bash
```

```bash
```

Function

```
bash (
```

```
bash (
```

```
bash (
```

```
bash (
```

```
bash (
```

```
bash (
```

```
bash (
```

```
bash (
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```
```

Adds the given address to the deny list of the specified type, preventing it from interacting with instances of that type as an input to a transaction. For coins, the type specified is the type of the coin, not the coin type itself. For example, "00...0123::my_coin::MY_COIN" would be the type, not "00...02::coin::Coin".

```bash
(
```

```bash
(
```

```bash
```

```bash
```

Removes a previously denied address from the list. Aborts with ENotDenied if the address is not on the list.

```bash
(
```

```bash
(
```

```bash
```

```bash
```

Returns true iff the given address is denied for the given type.

```bash
(
```

```bash
(
```

```bash
```

```bash
```

Creation of the deny list object is restricted to the system address via a system transaction.

```bash
```

```bash
```

```bash
```

```bash
```

Function

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Adds the given address to the deny list of the specified type, preventing it from interacting with instances of that type as an input to a transaction. For coins, the type specified is the type of the coin, not the coin type itself. For example, "00...0123::my_coin::MY_COIN" would be the type, not "00...02::coin::Coin".

```bash
(
```

```bash
(
```

```bash
```

```bash
```

Removes a previously denied address from the list. Aborts with ENotDenied if the address is not on the list.

```bash
(
```

```bash
(
```

```bash
```

```bash
```

Returns true iff the given address is denied for the given type.

```bash
(
```

```bash
(
```

```bash
```

```bash
```

Creation of the deny list object is restricted to the system address via a system transaction.

```bash
```

```bash
```

```bash
```

```bash
```

Function

```bash
(
```

```bash
(
```

```bash
(
```

```bash
(
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

Adds the given address to the deny list of the specified type, preventing it from interacting with instances of that type as an input to a transaction. For coins, the type specified is the type of the coin, not the coin type itself. For example, "00...0123::my_coin::MY_COIN" would be the type, not "00...02::coin::Coin".

```
bash (
```

```
bash (
```

```bash

```

```bash

```

Removes a previously denied address from the list. Aborts with [ENotDenied](#) if the address is not on the list.

```
bash (
```

```
bash (
```

```bash

```

```bash

```

Returns true iff the given address is denied for the given type.

```
bash (
```

```
bash (
```

```bash

```

```bash

```

Creation of the deny list object is restricted to the system address via a system transaction.

```bash

```

```bash

```

```bash

```

```bash

```

Function

```
bash (
```

```
bash (
```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

Adds the given address to the deny list of the specified type, preventing it from interacting with instances of that type as an input to a transaction. For coins, the type specified is the type of the coin, not the coin type itself. For example, "00...0123::my_coin::MY_COIN" would be the type, not "00...02::coin::Coin".

```bash
(
```

```bash
(
```

```bash
```

```bash
```

Removes a previously denied address from the list. Aborts with [ENotDenied](#) if the address is not on the list.

```bash
(
```

```bash
(
```

```bash
```

```bash
```

Returns true iff the given address is denied for the given type.

```bash
(
```

```bash
(
```

```bash
```

```bash
```

Creation of the deny list object is restricted to the system address via a system transaction.

```bash
```

```bash
```

```bash
```

```bash
```

Function

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Adds the given address to the deny list of the specified type, preventing it from interacting with instances of that type as an input to a transaction. For coins, the type specified is the type of the coin, not the coin type itself. For example, "00...0123::my_coin::MY_COIN" would be the type, not "00...02::coin::Coin".

```
bash (
```

```
bash (
```

```bash
```

```bash
```

Removes a previously denied address from the list. Aborts with ENotDenied if the address is not on the list.

```
bash (
```

```
bash (
```

```bash
```

```bash
```

Returns true iff the given address is denied for the given type.

```
bash (
```

```
bash (
```

```bash
```

```
```

```bash
```

Creation of the deny list object is restricted to the system address via a system transaction.

```bash
```

```bash
```

```bash
```

```bash
```

Function

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Adds the given address to the deny list of the specified type, preventing it from interacting with instances of that type as an input to a transaction. For coins, the type specified is the type of the coin, not the coin type itself. For example, "00...0123::my_coin::MY_COIN" would be the type, not "00...02::coin::Coin".

```
bash (
```

```
bash (
```

```bash
```

```bash
```

Removes a previously denied address from the list. Aborts with ENotDenied if the address is not on the list.

```
bash (
```

```
bash (
```

```bash
```

```bash
```

Returns true iff the given address is denied for the given type.

```
bash (
```

```
bash (
```

```bash
```

```bash
```

Creation of the deny list object is restricted to the system address via a system transaction.

```bash
```

```bash
```

```bash
```

```bash
```

Function

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Adds the given address to the deny list of the specified type, preventing it from interacting with instances of that type as an input to a transaction. For coins, the type specified is the type of the coin, not the coin type itself. For example, "00...0123::my_coin::MY_COIN" would be the type, not "00...02::coin::Coin".

```
bash (
```

```
bash (
```

```bash
```

```bash
```

Removes a previously denied address from the list. Aborts with ENotDenied if the address is not on the list.

```
bash (
```

```
bash (
```

```bash
```

```bash
```

Returns true iff the given address is denied for the given type.

```
bash (
```

```
bash (
```

```bash
```

```bash
```

Creation of the deny list object is restricted to the system address via a system transaction.

```bash
```

```bash
```

```bash
```

```bash
```

Function

```bash
```

```bash
```

```bash
```

```bash
```

Adds the given address to the deny list of the specified type, preventing it from interacting with instances of that type as an input to a transaction. For coins, the type specified is the type of the coin, not the coin type itself. For example, "00...0123::my_coin::MY_COIN" would be the type, not "00...02::coin::Coin".

```
bash (
```

```
bash (
```

```bash
```

```bash
```

Removes a previously denied address from the list. Aborts with [ENotDenied](#) if the address is not on the list.

```
bash (
```

```
bash (
```

```bash
```

```bash
```

Returns true iff the given address is denied for the given type.

```
bash (
```

```
bash (
```

```bash
```

```bash
```

Creation of the deny list object is restricted to the system address via a system transaction.

```bash
```

```bash
```

```bash
```

```bash
```

Macro function

```bash
```

```bash
```

Adds the given address to the deny list of the specified type, preventing it from interacting with instances of that type as an input to a transaction. For coins, the type specified is the type of the coin, not the coin type itself. For example, "00...0123::my_coin::MY_COIN" would be the type, not "00...02::coin::Coin".

```
bash (
```

```
bash (
```

```bash
```

```bash
```

Removes a previously denied address from the list. Aborts with [ENotDenied](#) if the address is not on the list.

```
bash (
```

```
bash (
```

```bash
```

```bash
```

Returns true iff the given address is denied for the given type.

```
bash (
```

```
bash (
```

```bash
```

```bash
```

```
```

Creation of the deny list object is restricted to the system address via a system transaction.

```bash

```

```bash

```

```bash

```

```bash

```

Function

Adds the given address to the deny list of the specified type, preventing it from interacting with instances of that type as an input to a transaction. For coins, the type specified is the type of the coin, not the coin type itself. For example, "00...0123::my_coin::MY_COIN" would be the type, not "00...02::coin::Coin".

```
bash (
```

```
bash (
```

```bash

```

```bash

```

Removes a previously denied address from the list. Aborts with [ENotDenied](#) if the address is not on the list.

```
bash (
```

```
bash (
```

```bash

```

```bash

```

Returns true iff the given address is denied for the given type.

```
bash (
```

```
bash (
```

```bash

```

```bash

```

Creation of the deny list object is restricted to the system address via a system transaction.

```bash
```

```bash
```

```bash
```

```bash
```

Function

```bash
```

```bash
```

Removes a previously denied address from the list. Aborts with [ENotDenied](#) if the address is not on the list.

```
bash (
```

```
bash (
```

```bash
```

```bash
```

Returns true iff the given address is denied for the given type.

```
bash (
```

```
bash (
```

```bash
```

```bash
```

Creation of the deny list object is restricted to the system address via a system transaction.

```bash
```

```bash
```

```bash
```

```bash

```

Function

Removes a previously denied address from the list. Aborts with [ENotDenied](#) if the address is not on the list.

```
bash (
```

```
bash (
```

```bash

```

```bash

```

Returns true iff the given address is denied for the given type.

```
bash (
```

```
bash (
```

```bash

```

```bash

```

Creation of the deny list object is restricted to the system address via a system transaction.

```bash

```

```bash

```

```bash

```

```bash

```

Function

```bash

```

```bash

```

Returns true iff the given address is denied for the given type.

```
bash (
```

```
bash (
```

```bash
```

```bash
```

Creation of the deny list object is restricted to the system address via a system transaction.

```bash
```

```bash
```

```bash
```

```bash
```

Function

Returns true iff the given address is denied for the given type.

```bash
bash (
```

```bash
bash (
```

```bash
```

```bash
```

Creation of the deny list object is restricted to the system address via a system transaction.

```bash
```

```bash
```

```bash
```

```bash
```

Function

```bash
```

```bash
```

Creation of the deny list object is restricted to the system address via a system transaction.

```bash
```

```bash
```

```bash
```

```bash
```

Function

Creation of the deny list object is restricted to the system address via a system transaction.

```bash
```

```bash
```

```bash
```

```bash
```

Function

```bash
```

```bash
```