Address types

In the Filecoin network, an address is a unique identifier that refers to an actor in the Filecoin state. All actors in Filecoin have a corresponding address which varies from the different usages.

Filecoin has five address classes, and actors tend to havemultiple addresses. Furthermore, each address class has its own rules for converting between binary and text.

The goal of using different types of addresses is to provide a robust address format that is scalable, easy to use, and reliable. These addresses encode information including：

- Network prefix: indicates the network the actor belongs to.
- Protocol indicator: identify the type and version of this address.
- Payload: identify the actor according to the protocol.
- Checksum: validate the address.
-

Filecoin addresses can be represented either as raw bytes or a string. Raw bytes format will always be used on-chain. An address can also be encoded to a string, including a checksum and network prefix. The string format will never appear on-chain and is only for human-readable purposes.

Filecoin address can be broken down like this:

Network prefix Protocol indicator Payload Checksum f /t 1 byte:0 /1 /2 /3 /4 n bytes 4 bytes The network prefix is prepended to an address when encoding to a string. The network prefix indicates which network an address belongs to. Network prefixes never appear on-chain and are only used when encoding an address to a human-readable format.

- f
-
    - addresses on the Filecoin mainnet.
- t
-
    - addresses used on any Filecoin testnet.
-

The protocol indicator identifies the address type, which describes how a method should interpret the information in thepayload field of an address.

- 0
- : An ID address.
- 1
- : A wallet address generated from a secp256k public key.
- 2
- : An actor address.
- 3
- : A wallet address generated from BLS public key.
- 4
- : A delegated address for user-defined foreign actors:
-
    - 410
-
    - : Ethereum-compatible address space managed by the Ethereum address manager (EAM). Each 410 address is equivalent to an 0x address.
- *
-

Each address type is described below.

ID addresses

All addresses have a short integer assigned to them byInitActor sequentially, a unique actor that can createnew actors. The integer that gets assigned is the ID of that actor. AnID address is an actor’s ID prefixed with the network identifier and the protocol indicator. Therefore, any address in the Filecoin network has a unique ID address assigned to it.

The mainnet burn account ID address isf099 and is structured as follows:

```

Copy Protocol Indicator | f 0 9 9 | | | Actor ID | Network identifier
```

```

Actor addresses

Addressed representing an actor deployed through the init actor in the Filecoin network. It provides a way to create robust addresses for actors not associated with a public key. They are generated by taking asha256 hash of the output of the account creation.

Actor addresses are often referred to by their shorthand,2 .

Wallet addresses

Addresses managed directly by users, like accounts, are derived from a public-private key pair. If you have access to a private key, you can sign messages sent from that wallet address. The public key is used to derive an address for the actor. Public key addresses are referred to asrobust addresses as they do not depend on the Filecoin chain state.

Public key addresses allow devices, like hardware wallets, to derive a valid Filecoin address for your account using just the public key. The device doesn't need to ask a remote node what your ID address is. Public key addresses provide a concise, safe, human-readable way to reference actors before the chain state is final. ID addresses are a space-efficient way to identify actors in the Filecoin chain state, where every byte matters.

Filecoin supports two types of public key addresses:

- secp256k1 addresses
- that begin with the protocol indicator as1
- .
- BLS addresses
- that begin with the protocol indicator as3
- .
-

t1iandfn6d...ddboqxbhoeva - a testnet wallet address generated using secp256k1.t3vxj34sbdr3...road7cbygq - a testnet wallet address generated using BLS.

Delegated addresses

Filecoin supports extensible, user-defined actor addresses through the4 address class, introduced in Filecoin Improvement Proposal (FIP) 0048 . The4 address class provides the following benefits to the network:

- Implement foreign addressing systems in Filecoin.
- A predictable addressing scheme to support interactions with addresses that do not yet exist on-chain.
- User-defined, programmable addressing systems without extensive changes and network upgrades.
-

For example, a testnet delegated address using the Ethereum Addressing System is structured as follows:

```

Copy Address manager actor ID | t 410 iandfn6d... | | | New actor ID | Network identifier

```

Theaddress manager actor ID is the actor ID of the address manager actor, which creates new actors and assigns a4 address to the new actor. This leverages the extensible feature of thef4 address class.

Thenew actor ID is the arbitrary actor ID chosen by that actor.

Restrictions

Currently, perFIP 0048 ,f4 addresses may only be assigned by and in association with specific, built-in actors calledaddress managers . This restriction will likely be relaxed once users are able to deploy custom WebAssembly actors.

This address type plays an essential role in supporting the FEVM. It allows the Filecoin network to be able to recognize the foreign address and validate and execute the transactions sent and signed by the supported foreign addresses.

The supported foreign addresses can be cast asf4/t4 addresses, and vice-versa. But not withf1/t1 orf3/t3 addresses.

Ethereum Address Manager

Ethereum Address Manager (EAM) is a built-in actor that manages the Ethereum address space, anchored at the410 address namespace. It acts like an EVM smart contract factory, offering methods to create and assign thef410/t410 Filecoin address to Ethereum address.
```

The subaddress of anf410/t410 address is the original Ethereum address. Ethereum addresses can be cast asf410 addresses, and vice-versa. Thef410/t410 address will be used for the Ethereum-compatible FVM (FEVM) development tools and applications built on FEVM.

Example

```

Copy

# An Ethereum wallet address.

0xd388ab098ed3e84c0d808776440b48f685198498

# The corresponding Filecoin address on Calibration.

t410f2oekwcmo2pueydmaq53eic2i62crtbeyuzx2gmy

```

If you have an Ethereum wallet address starting with0x , then the Ethereum Address Manager (EAM) will assign a correspondingt410 Filecoin address to it. If you send 10 TFIL to0xd388ab098ed3e84c0d808776440b48f685198498 using a wallet like MetaMask, you will receive 10 TFIL to yourt410f2oekwcmo2pueydmaq53eic2i62crtbeyuzx2gmy address on Filecoin Calibration testnet.

```

Copy

# A Filecoin smart contract address.

t410fl5qeigmkcytz7b6sqoojtcetqwf37dm4zv4aijq

# The corresponding Ethereum smart contract address.

0x5f6044198a16279f87d2839c998893858bbf8d9c

```

Again, assume you have deployed a solidity smart contract on Filecoin Calibration. Then you will receive a smart contract address starting witht410 . EAM will also assign a corresponding0x Ethereum address to it.

When you try to invoke this smart contract on Filecoin using Ethereum tooling, you need to use your0x5f6044198a16279f87d2839c998893858bbf8d9c smart contract address.

Converting to a 0x-style address

The Filecoin EVM runtime introduces support for0x Ethereum-style addresses. Filecoin addresses starting with eitherf0 orf410f can be converted to the0x format as follows:

Addresses starting withf0 can be converted to the0x format by:

- Extracting theactor_id
- (e.g., the1234
- inf01234
-).
- Hex encode with a0xff
- prefix:sprintf("0xff0000000000000000000000%016x", actor_id)
- .
-

Addresses starting withf410f address can be converted to the0x format by:

- Removing thef410f
- prefix.

- Decoding the remainder as base 32 (RFC 4648 without padding).
- Trim off the last 4 bytes. This is achecksum
- that can optionally be verified, but that’s beyond the scope of this documentation.
- Assert that the remaining address is 20 bytes long.
- Hex-encode:sprintf(0x%040x", actor_id)
- .
-

f0 addresses arenot re-org stable and should not be used until the chain has settled.

Previous Actor types Next FILForwarder

Last updated7 months ago