Precompiles

A precompile refers to a pre-existing piece of code or a smart contract that is already deployed on the Filecoin network for use by developers.

The Filecoin virtual machine (FVM) has several pre-compiled contracts called precompiles. Each precompile address starts with0xfe000... . Specifically:

- Resolve address0xfe00..01
- Lookup delegated address0xfe00..02
- Call actor by address0xfe00..03
- Call actor by ID0xfe00..05
-

Resolve Address

Address:0xfe0000000000000000000000000000000000000001

Resolves a Filecoin address (e.g., “f01”, “f2abcde”) into a Filecoin actor ID (uint64). Every actor in Filecoin has an actor ID.

- Input: The Filecoin address in itsbytes
- representation.
- Output:
-
  - If the target actor exists, succeed and return an ABI-encoded actor ID (u64).
-
  - If the target actor doesn’t exist, succeed with no return value.
-
  - If the supplied address is invalid (cannot be parsed as a Filecoin address), revert.
- *
-

Example:

```

Copy (boolsuccess,bytesmemoryactor_id_bytes)=address(0xfe0000000000000000000000000000000000000001).staticcall(fil_address_bytes); require(success,"invalid address"); require(actor_id_bytes.length==32,"actor not found"); uint64actor_id=abi.decode(actor_id_bytes);

```

Lookup Delegated Address

Address:0xfe0000000000000000000000000000000000000002

Looks up the “delegated address” (f4 address) of an actor by ID. This precompile isusually used to lookup the Ethereum-style address of an actor by:

1. Looking up the delegated address.
2. Checking that the delegated address is 22 bytes long and starts with0x040a
3. .
4. Returning the last 20 bytes (which will be the Ethereum-style address of the target actor).
5.
6. Input: An ABI-encoded actor ID (u64 encoded as a u256).
7. Output:
8.
   - If the supplied actor ID is larger than max u64, revert.
9.
   - If the target actor exists and has a delegated address, succeed and return the delegated address as raw bytes.
10.
   - Otherwise, succeed with no return value.
11. *
12.

Example:

```

Copy (boolsuccess,bytesmemorydelegated_address_bytes)=address(0xfe0000000000000000000000000000000000000002).staticcall(abi.encode(uint256(actor_id)));

```

Call Actor By Address

Address:0xfe0000000000000000000000000000000000000003

Calls the specified actor using the native FVM calling convention by itsFilecoin address. This precompile must be called withDELEGATECALL as the precompile will call the target actoron behalf of the currently executing contract.

Input: ABI Encoded

```

Copy (uint64method, uint256value, uint64flags, uint64codec, bytes params, bytes filAddress)

```

- method
- is the Filecoin method number. The precompile will revert if the method number is not either 0 (bare value transfer) or at least 1024. Methods between 1 and 1023 inclusive are currently restricted (but may be allowed in the future).
- value
- is the value to transfer in attoFIL.
- codec
- is the IPLD codec of the parameters. This must either be 0x51 or 0x00 (for now) and will revert if passed an illegal codec:
-

- If the parameters are non-empty, they must be CBOR, and the codec must be 0x51.
- If the parameters are empty, the codec must be 0x00.
- *
- params
- are the CBOR-encoded message parameters, if any.
- filAddress
- is the Filecoin address of the caller.
-

Output: ABI Encoded

```

Copy (int256 exit_code, uint64 return_codec, bytes return_value)

```

- exit_code
- is one of:
- = 0
- to indicate the call exited successfully.
- 0
- to indicate that the target actorreverted
- with the specifiedexit_code
- .
- < 0
- to indicate the call itself failed with thesyscall-error
- -exit_code
- .
- *
- return_codec
- codec of returned data. This will be one of (for now):
- 0x51 or 0x71 - CBOR
- 0x55 - raw (the target actor returned raw data)
- 0x00 - nothing (the returned data will be empty as well).
- *
-

This precompile only reverts if an input is statically invalid. If the precompile fails to call the target actor for any other reason, it will return a non-zeroexit_code but will not revert. Example:

```

Copy
(boolsuccess,bytesmemorydata)=address(0xfe0000000000000000000000000000000000000003).delegatecall(abi.encode(method,value,flags,codec,params,filAddress));
(int256exit,uint64return_codec,bytesmemoryreturn_value)=abi.decode(data,(int256,uint64,bytes));

```

Call Actor By ID

Address:0xfe0000000000000000000000000000000000000005

This precompile is identical to the “Call Actor By Address” (0xfe00..03) except that it accepts an actor ID (uint64) instead of an actor address as the last parameter. That is:

```

Copy (uint64method,uint256value,uint64flags,uint64codec,bytesparams,uint64actorId)

```

Example:

```

Copy (boolsuccess,bytesmemorydata)=address(0xfe0000000000000000000000000000000000000005).delegatecall(abi.encode(method,value,flags,codec,params,id));
(int256exit,uint64return_codec,bytesmemoryreturn_value)=abi.deco

```

Previous How gas works Next Programmatic storage

Last updated10 days ago