Conflux eSpace

Conflux Network   v2.0   (   ) - eSpace   RPC   EVM   e

- [eSpace Quick Start](#)
- [Spaces](#)
- [CrossSpaceCall](#)
- [CheatSheet](#)
- [FAQs](#)
- [EN Translate](#)
    - [Spaces](#)

eSpace Quick Start

Metamask

CFX

Hardhat   ERC20 Token

Note

Spaces

Space

Conflux v2.0(Hydra) IP-90 `eSpace`, RPC EVM `CoreSpace`

`Conflux TX`

Core eSpace eSpace Core Avalanche

eSpace

eSpace

Conflux Conflux

eSpace SDK eSpace

Space

CoreSpace eSpace

- Core Conflux (Fluent), SDK(*-conflux-sdk), (chainIDE, conflux-truffle)
- eSpace Metamask, truffle, hardhat, ethers.js RPC Conflux

Space

Space

eSpace `CrossSpaceCall` CFX

(Mirror Address)

Core `Mirror Address`, eSpace Core Core eSpace

1. Base32 decode Buffer payload
2. payload keccak 20
3. hex eSpace

Conflux SDK base32

```
> const { address } = require('js-conflux-sdk')
undefined
> address.cfxMappedEVMSpaceAddress('cfx: aamgvyzht7h1zxdghb9ee9w26wrz8rd3gj837392dp')
'0x62954816cE133B41Ab888e1b68b62549DE2f32e0'
>
```

base32 decode hex eSpace

CrossSpaceCall

CrossSpaceCall `Core` 0x0888000000000000000000000000000000000006

CrossSpaceCall

1. CFX
2. Core eSpace
3. Core eSpace

CFX [ConfluxHub CrossSpace app](#) CFX ERC20

CrossSpaceCall

Conflux Core ⬜ TPS Base32 RPC, SDK Core gas

Conflux eSpace eSpace

- [CIP-90](#)
- [RPC Compatibility](#)
- [EVM Compatibility](#)

CrossSpaceCall

CheatSheet

RPC & chainId

Mainnet

- RPC: https://evm.confluxrpc.com
- ChainId: 1030

Testnet

- RPC: https://evm.confluxrpc.org
- ChainId: 71

Scan

- https://evm.confluxscan.io/

Testnet Faucet

- https://efaucet.confluxnetwork.org/

SpaceBridge

- https://confluxhub.io/

L1 bridges

- Multichain
- cBridge
- Meson

FAQs

eSpace

Core Core

eSpace

eSpace 1 1s 40-50

eSpace gasLimit

eSpace gasLimit 1500w gas gas CFX 21000 gas 700

eSpace 5 eSpace 600w gas .

eSpace 1559

155 legacy

eSpace PoW or PoS

PoW

eSpace Core RPC ?

RPC 50-100

eSpace standardV

standardV V

eSpace Scan gasPrice 0

CrossSpaceCall Phantom Transaction gasPrice 0

EN Translate

Spaces

The Concept of Space

In Conflux v2.0 (Hydra) upgrade, a new space called `eSpace` is introduced by [CIP-90](#). It is an `independent space (or chain)` which is compatible with Ethereum `interface`, including `RPC` and `EVM`. The original chain will be called `CoreSpace` and remain unchanged and unaffected.

Space is an abstract concept that is used to isolate and distinguish `Ethereum format transaction` from `Conflux transaction`. The two spaces are logically independent, and they do not affect each other.

To better understand, you can think CoreSpace and eSpace as two separate chains, where eSpace is a sub-chain of CoreSpace.

In other ways, we can refer to the `virtualization technology` from operating system concepts, and hence understood eSpace as a `virtualized` Ethereum chain running in the original network.

Why Introduce eSpace?

Conflux is a high-performance, fully decentralized public chain enabled by an innovative Tree-Graph consensus algorithm. The transaction fee of Conflux is very low that can be seen as almost free compared to other networks such as Ethereum. But consider Ethereum has already build a mature ecosystem including tools, SDKs, wallets, Solidity libraries. In order to reduce the migration cost of projects and users, and make users experience the advantages of low fees and high TPS of Conflux, eSpace was introduced.

Through the fully compatible interface, smart contracts and dApps of Ethereum can be directly deployed to eSpace without any modification. Development tools, SDKs, wallets, and services of Ethereum can be directly used in eSpace. Users do not need to learn new knowledge but just use the original tools to get started directly.

eSpace is very easy to use for ethereum developers and users, just like BSC, Polygon, Aurora.

The Relationship Between the Two Space

CoreSpace and eSpace are two logically independent spaces with their own transactions, account status, and contracts.

- To interact with CoreSpace use Conflux-compatible wallet (Fluent), SDK (*-conflux-SDK), and development tools (chainIDE, conflux-truffle).
- To interaction with eSpace directly use the existing tools and products from the Ethereum ecosystem, such as Metamask, truffle, hardhat, ethers.js, etc. (by simply setting the RPC network of the tool to Conflux eSpace RPC).

In terms of the underlying data storage, both Spaces reuse the same ledger (chain). A block may contain transactions from both Spaces, and they are only differentiated by the transaction type when transactions are executed. Each will only impact the account status in its own Space.

How to Communicate Between Spaces

The brand new eSpace chain has not introduced new tokens. CFX is still used as native token in eSpace to pay transaction fees. Which can be transferred across the two spaces by the newly introduced internal contract `CrossSpaceCall`.

Mirror Address

Each account in CoreSpace has a corresponding account in eSpace, which we call the `Mirror Address`. However, eSpace accounts do not have the corresponding account in CoreSpace. The mirror address from CoreSpace to eSpace is calculated by fixed steps:

1. Decode the original Base32 address to get the address payload with the Buffer format.
2. Hash the payload by Keccak and take the last 20 bytes
3. The mirror address is the result of the previous step in hex encoding

Conflux SDKs provide methods to calculate the base32 mirror address.

```
// js-conflux-sdk example
> const { address } = require('js-conflux-sdk')
```

```
> address.cfxMappedEVMSpaceAddress('cfx:aamgvyzht7h1zxdghb9ee9w26wrz8rd3gj837392dp')
'0x62954816cE133B41Ab888e1b68b62549DE2f32e0'
```

Note: Do not directly use the hex address that is resulted from decoding the base32 address as the `eSpace mirror address` or `eSpace address`. This operation is wrong and will lead to loss of assets.

Internal Contract CrossSpaceCall

CrossSpaceCall is an internal contract located in `CoreSpace` with hex40 format address `0x0888000000000000000000000000000000000006`. The specific interface of the contract can be found in its [interface documentation](#)

The internal contract CrossSpaceCall provides several functions.

1. It enables the cross-space transfers of CFX.
2. It enables deploy contracts to eSpace from CoreSpace.
3. It enables eSpace contract methods to be called in CoreSpace.

The cross-space transfers of CFX are synchronous and can be done within one transaction. It is simpler, safer, and faster compared to cross-chain transfers.

[ConfluxHub CrossSpace app](#) is a CFX, ERC20 cross-space transfer application based on this internal contract, which is very user-friendly.

The detail of CrossSpaceCall can be found in the [separate section](#).

How To Choose

Conflux CoreSpace is a native space that supports `contract sponsorship` and `has more network capacity (higher TPS)`, but requires unique Base32 addresses, RPC, SDK, etc. Therefore, if you want to develop a brand new project, you can choose the CoreSpace and the users of the project can interact with the project by paying no gas fee.

If you want to deploy an Ethereum project to take advantage of Conflux's high performance and low cost in order to reduce user costs, you can choose eSpace. The project can be deployed directly without any modification. If you are a skilled Ethereum engineer, you can also choose eSpace directly and use the tools and SDKs that you are familiar with to get started quickly.

Reference

- CIP-90
- RPC Compatibility
- EVM Compatibility

- CIP-90
- RPC Compatibility
- EVM Compatibility