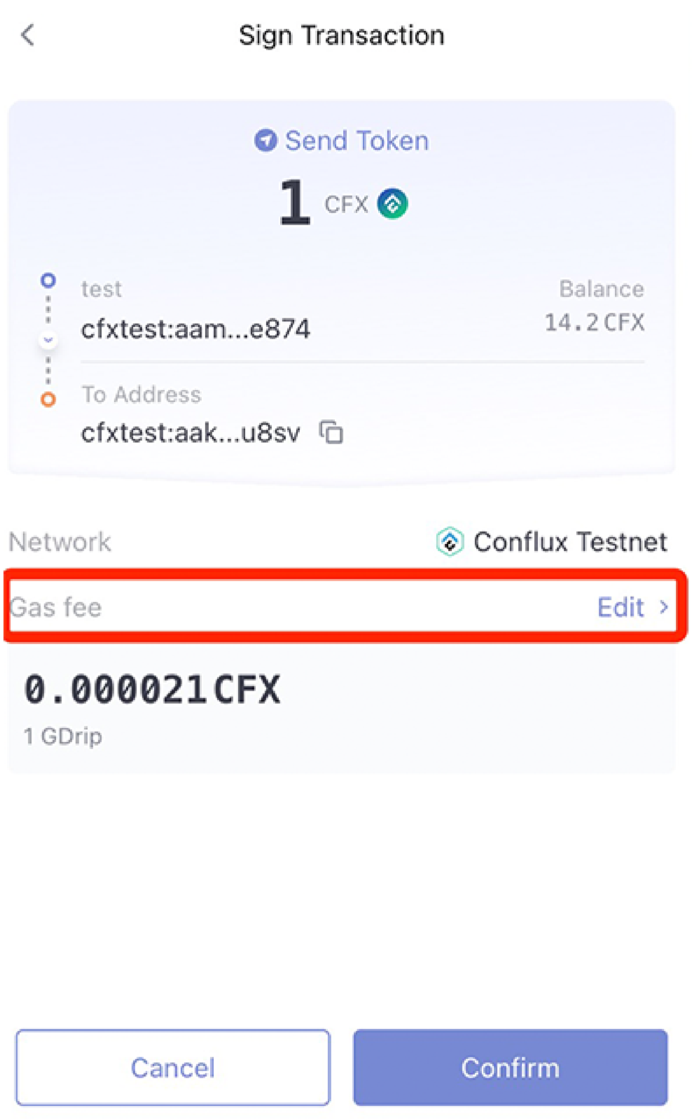
EN Translate

- Gas
- Accounts

Gas

Conflux users usually see fields like `gasFee`, `gas`, and `gasPrice` when they are sending transactions using their wallets (Fluent) or SDK. This article is going to explain in detail about what these concepts mean and how to set the values properly.



Sign Transaction

Send Token

1 CFX

test

cfxtest:aam...e874

Balance 14.2 CFX

To Address

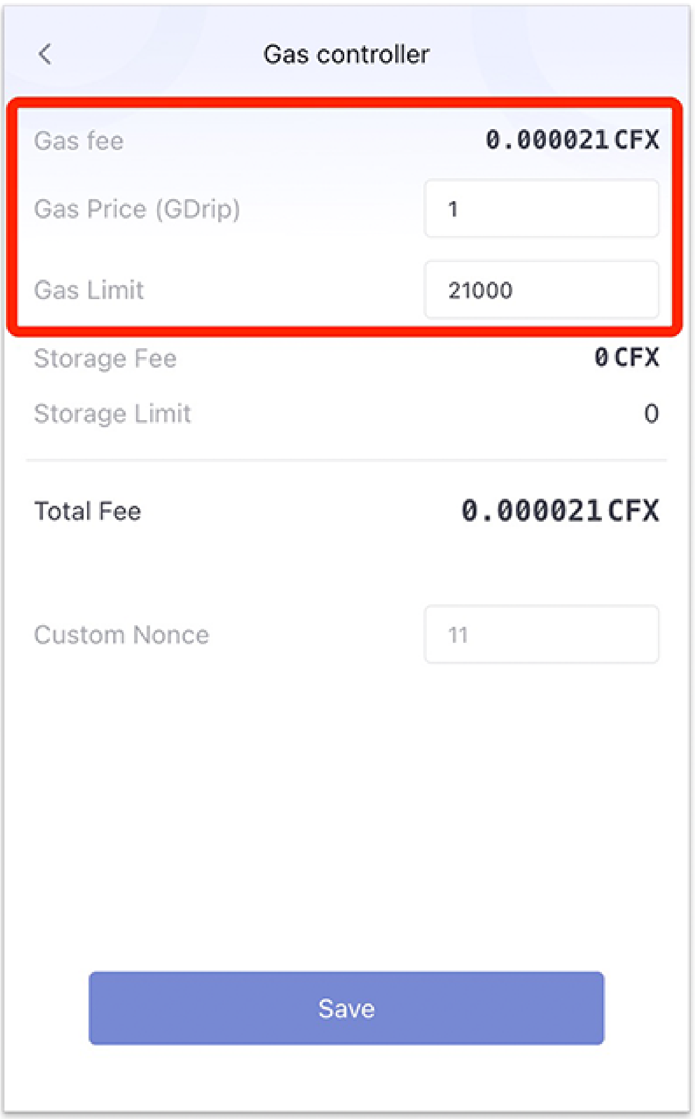
cfxtest:aak...u8sv

Network: Conflux Testnet

Gas fee: Edit

0.000021 CFX

1 GDrip

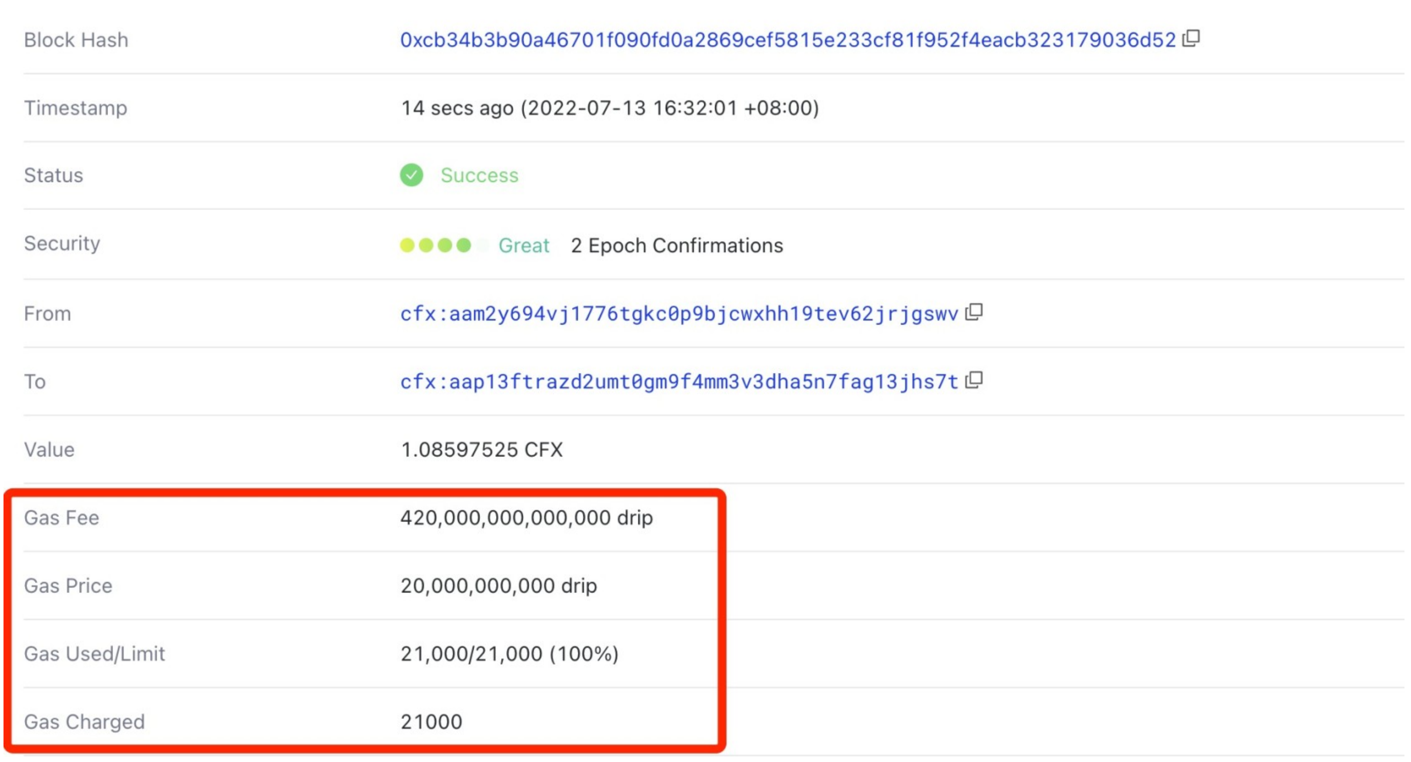


Gas controller

Gas fee	0.000021 CFX
Gas Price (GDrip)	1
Gas Limit	21000
Storage Fee	0 CFX
Storage Limit	0
Total Fee	0.000021 CFX
Custom Nonce	11

gasFee

In real life, when we send money to someone else at a bank, we usually have to pay transaction fees. It is the same for sending transactions in blockchains (Bitcoin, Ethereum, Conflux). `gasFee` is the fee for sending a transaction. Usually, it needs to be paid by the native token of the chain. Take Conflux as an example, CFX is needed for paying the transaction fee (gas fee).



Block Hash	0xcb34b3b90a46701f090fd0a2869cef5815e233cf81f952f4eacb323179036d52
Timestamp	14 secs ago (2022-07-13 16:32:01 +08:00)
Status	Success
Security	●●●● Great 2 Epoch Confirmations
From	cfx:aam2y694vj1776tgkc0p9bjcwxhh19tev62jrjgswv
To	cfx:aap13ftrazd2umt0gm9f4mm3v3dha5n7fag13jhs7t
Value	1.08597525 CFX
Gas Fee	420,000,000,000,000 drip
Gas Price	20,000,000,000 drip
Gas Used/Limit	21,000/21,000 (100%)
Gas Charged	21000

Why Pay Fees

As we all know, blockchain is in fact a decentralized ledger, which is maintained by miners. There is a cost for miners to store data and generate blocks (calculating hashes). Therefore, in order to motivate miners to actively participate in the chain maintenance and protect the network security, the blockchain consensus system is designed to include a reward mechanism for miners, and the transaction fee is one of the rewards for miners, which will be paid to miners who participate in generating blocks. This mechanism can ensure the sustainability of the entire decentralized network.

In addition, the gas fee mechanism can also prevent abusive transactions and thus improve the efficiency of blockchain utilization.

What is Gas

The concept of gas is borrowed from the real 'gas', petrol. Vehicles consume gasoline to drive. The further a car travels, the more gasoline it consumes. In EVM blockchains, gas represents the work amount required to execute a transaction. Therefore, it is a unit that measures the amount of computation required to perform certain operations.

To provide more details, all Conflux transactions are executed by EVM, including regular CFX transfers and smart contract method calls. When these operations are executed, they are compiled into individual OPCodes. The amount of work required to execute each OPCode varies. More information for OPCode gas fees can be found here.

Usually, the gas consumed for a regular CFX transfer is `21000`. A smart contract transaction usually needs more, depending on the complexity of the contract execution.

Gas Limit

When constructing a transaction, the `gas` field is very important, as the field itself means the `maximum amount of gas` that can be consumed by the execution of the transaction.

It is very important to fill the gas field correctly. If the gas limit is set to a value less than the actual amount of gas needed, the transaction will fail. If the gas limit is set too high, you may pay more gas than you actually need to.

The maximum gas limit for a single transaction in the Conflux network is 15M. This ensures that the transactions will not overconsume EVM resources.

gasPrice

The gasPrice of a transaction is specified by the sender of the transaction and represents the fee that the person is willing to pay per unit of gas. The unit of gasPrice is GDrip, where 1 GDrip is equal to 0.000000001 CFX (10^{-9} CFX).

A transaction's gasPrice value affects how fast the transaction is packaged by miners, as miners will prioritize packaging transactions with higher gasPrice in order to make more profits. When the network is not congested, setting gasPrice to `1Gdrip` is enough to be packed normally. However, when the network is congested, more transactions are waiting to be packed. At this time, if the gasPrice is set to be less than most other transactions, it will not be packed but keep waiting.

Therefore, if you want the transaction to be packaged quickly, you can set the gasPrice higher. Usually setting it to 10G-1000G is high enough in Conflux to ensure it is executed quickly.

NOTE: Do not set gasPrice too high. It may lead to sky-high transaction fees. If gasPrice is set to 1CFX, then the fee for a regular transfer is 21000 CFX, which is quite a lot for a transaction.

How gasFee is Calculated

gasFee is the total gas fee paid for a transaction. It is calculated as `gasFee = gasUsed * gasPrice`. gasFee takes the smallest unit of CFX, Drip.

Suppose there is a regular transfer of 1 CFX, the gas limit can be set to 21,000. If the gasPrice is set to 1GDrip, then the total cost of the transaction is `1 + 21000 * 0.000000001 = 1.000021 CFX`, where 1 CFX is transferred to the recipient's account, and 0.000021 CFX is the reward for the miner.

In addition, in a Conflux transaction, if the `gas limit` is more than the actual amount of gas consumed (`gasUsed`), the exceeding part will be returned. The returning amount of gas can only be up to a quarter of the `gas limit`.

Suppose the gas limit for a regular CFX transfer is set to 100k and the actual execution consumed 21,000, since the gas limit for the transaction is set too high, at most 25,000 of the gas fee will be returned(25% of the gas limit). The gas used for the transaction will be `0.000075 CFX.`

If the gas limit setting of the transaction is not that high, take the same example as above but set the gas limit to 25000, which is 4000 more than the actual amount used, the exceeding part is not more than a quarter of the gas limit. This part will be returned fully, and the final amount of fees charged will still be `0.000021 CFX`.

How to Set gas and gasPrice Properly

gasPrice

Usually, gasPrice can be set within the range of 1G to 10,000G. If the network is not congested, 1G would be enough. If the waiting time after sending the transaction is too long, you can set the gasPrice to 10G or 100G.

gas

For regular CFX transfers, setting the gas to 21,000 is enough.

For contract interactions, you can estimate the gas value by using the method `cfx_estimateGasAndCollateral`, which simulates the execution of the transaction and returns the actual amount of gas used for the transaction. In most cases the value returned by this method is accurate, but sometimes there could be underestimations. A safe way of dealing with this is to multiply the result of the estimation by a factor `1.3`. This ensures that the gas limit is sufficient for the transaction, and any excessive gas fee will be refunded.

FAQs

1. Are there any EIP-1559 transactions in the Conflux network?

Currently, in the Conflux network, there are only transactions that correspond to the EIP-155 standard.

Further Readings

- Ethereum Developer Documentation: Gas and Fees
- Ethereum Gas Explained

Accounts

`Account` is a very important object entity in the Conflux network. It is used to store CFX (every account has its CFX balance) and send Conflux transactions. Accounts and account balances are stored in a huge table in the Conflux VM, and they are part of the full state of the Conflux ledger.

Types of Accounts

Conflux has two types of accounts.

- External accounts (private key accounts) - are controlled by the holder of the private key
- Smart Contracts - are the ones deployed in the network and controlled by the contract codes

Note: There is a special type of smart contract in the Conflux network - the internal contracts. They are created automatically when the network is started or upgraded, but not by deploying contract codes. There are currently 6 internal contracts.

Similarities of Accounts

- Both of them can accept, hold, and send CFX and tokens
- Both of them can interact with smart contracts in the network

Differences of Accounts

External Accounts

- Creating external accounts does not have costs, such as CFX or other resources
- They can send transactions to others
- Transactions between external accounts can only be CFX or token transactions

Smart Contracts

- Creating smart contracts does have costs, as it uses the network's storage and computational resources
- Transactions can only be sent to other contracts as a response to a received transaction
- Transactions sent from external accounts to contract accounts can trigger the smart contract's codes to perform many different operations, such as token transfers, creating new contracts, etc.

External Accounts and Public-private Key Pairs

An external account is generated by a public-private key pair that can be used to prove that a transaction was indeed signed by the account, in order to prevent transaction falsification. The private key is used by the user to sign a transaction. It gives you the right to operate the assets on the account corresponding to the private key. Essentially, the user does not hold the CFX or the tokens but the private key. CFX is always present in Conflux's ledger.

This mechanism prevents malicious users from broadcasting fake transactions, as we can verify the address that sends the transaction at any time.

For example, if Alice wants to send CFX from her account to Bob's account, she needs to create a transaction and send it to the network for verification. Conflux uses the public key encryption mechanism to ensure that Alice can prove that the transaction is sent by herself. Suppose now, Eve, a malicious user, directly broadcasts a transaction, say, "send 5 CFX from Alice's account to Eve's account". Without the above-mentioned mechanism, no one would be able to verify that the transaction was sent by Alice.

External Account Creation

When you want to create an external account, you can use a wallet, like FluentWallet, or any language library, where essentially both of them generate a random private key.

A private key contains 64 hexadecimal characters and can be encrypted using a password.

```
fffffffffffffffffffffffffffffffebaaedce6af48a03bbfd25e8cd036415f
```

The public key is calculated from the private key by the Elliptic Curve Cryptography Algorithm. Then, Keccak-256 hashing is performed on the public key, and the Conflux address is yielded by encoding the last 20 bytes (the first 4 bit will be set to 0001) of the result with base 32 formats.

```
// Mainnet address
cfx:aatktb2te25ub7dmyag3p8bbdgr31vrbeackztm2rj
// Testnet address
cfxtest:aatktb2te25ub7dmyag3p8bbdgr31vrbeajcg9pwkc
```

The public key can be calculated from the private key, but the private key cannot be calculated from the public key. The private key has to be kept safe by the user.

Smart Contract Account

Smart contracts also have base32 encoded addresses

```
cfx:acf2rcsh8payyxpg6xj7b0ztswwh81ute60tsw35j7
```

This address is determined when the contract is deployed and is calculated by the deployed transaction's `sender address`, `nonce`, and the smart contract's `code`.

Note: The addresses of internal contracts are special - they are assigned by the network itself.

Details of Accounts

The global state of Conflux is composed of individual account states, each of which is an address-state pair (key pair).

A Conflux account state includes five parts:

- `Basic state` is the basic state of the account.
- `Storage state` is a key/value database or storage space that can be used to store custom states or data of smart contracts.
- `Code information` is the code information of the smart contract account. It includes the `contract codes` and the `address` of the account that paid the fee for the storage space occupied by the codes.
- `Staking deposit list` is the list of Staking operations of the accounts (it will be removed in the next Hardfork).
- `Staking vote lock list` is the list of lock operations performed by the account to participate in DAO voting.

The basic status of the account consists of eight fields as follows:

- `nonce` is a counter to keep track of the number of transactions sent by an account. It is also used to ensure that each transaction can only be executed once. For contract accounts, this value indicates the number of `contracts created by this contract`.
- `balance` is the number of CFX of the address in Drip. Drip is the smallest unit of CFX, where 1CFX=10^{18} Drip.
- `codeHash` is the hash of the code of the contract account. The user can reference the contract code, the code cannot be modified after the contract is created. The code will be executed when the contract receives a message call. For external accounts, codeHash is a hash of an empty string.
- `stakingBalance` is the balance of the staked amount. Similarly, the unit is Drip.
- `admin` is the administrator address of the `contract account` recorded in the AdminControl internal contract. In default, the contract administrator is set to the account which deployed this contract when it is created. The administrator can destroy the contract it

manages through the internal contract AdminControl, or give the administrator role to another account. The administrator address of an external account is itself.

- `sponsorInfo` is the information of the contract sponsor. It contains `sponsor for gas`, `sponsor for collateral`, `sponsor gas limit`, `sponsor balance for gas`, and `sponsor balance for collateral`.
- `storageCollateral` is the amount of Drip staked to use the storage spaces.
- `accumulatedInterestReturn` is the amount of cumulative reward of the account from Staking. The unit of it is Drip. Starting with Conflux 2.0, users must also participate in PoS in order to receive the reward.

For more details about accounts, please refer to the `Accounts` section in Conflux Protocol Specification.