

# Fees on Namada

In order to settle the market for Namada blockspace demand, fees are coupled with transactions. In order for any namada transaction to be considered valid, the correct corresponding fee must be paid. The exact fee is set by the user, and must be greater than or equal to the minimumgas-price set by governance, which is included in the genesis file undergas\_cost .

## How fees are paid

When explicitly stated, the gas fee is paid by the--gas-payer flag. If no--gas-payer flag is specified, the gas fee is paid by the first key in the--signing-keys flag.

ðŹ Note the use of the placeholderkeysha for the key parameter. This is a completely configurable parameter, and should just refer to the alias of the key signing the transaction (that has a positive nam balance). This means that in the transaction

```
namada
```

```
client
```

```
transfer \ --source
```

```
my-new-acc \ --target
```

```
validator-1 \ --token
```

```
NAM \ --amount
```

```
10 \ --signing-keys
```

keysha the account associated with keykeysha will be required to pay the fee. This means that even though the accountaccountant may have a positive NAM balance,keysha will need to have the associated NAM in order to pay the transaction fee.

A user can also specify a--gas-payer flag to specify a different account to pay the fee. This is useful in cases where the account that is signing the transaction does not have enough NAM to pay the fee. For example, ifkeysha has a balance of 5 NAM, butaccountant has a balance of 100 NAM, then the following transaction will be valid:

```
namada
```

```
client
```

```
transfer \ --source
```

```
my-new-acc \ --target
```

```
validator-1 \ --token
```

```
NAM \ --amount
```

```
10 \ --signing-keys
```

```
keysha \ --gas-payer
```

accountant (Assuming thataccountant exists and is in the wallet of the user)

For testnet purposes, we recommend[using the faucet](#) to source NAM for transaction fees.

## How fees are calculated

The fee for a transaction is calculated by multiplyinggas-limit by the gas price. Both the--gas-limit and the--gas-price can be specified by the user. If neither is specified, the default gas limit and minimum gas price is used. The default gas limit for any transaction is currently set to20\_000 .

The minimum gas price is set in the genesis file undergas\_cost .

## How to set the gas price and gas limit

It is recommended to set thegas-limit based on the transaction being conducted. In order to estimate thegas-limit for a particular transaction, the--dry-run-wrapper argument should be provided. This means that the transaction is simulated, but not yet sent to the ledger.

For example, the following command will simulate a transfer transaction, and return the gas used:

```
namadac
```

```
transfer \ --source
```

```
my-new-acc \ --target
```

```
validator-1 \ --token
```

```
NAM \ --amount
```

```
10 \ --signing-keys
```

```
keysha \ --gas-payer
```

```
accountant \ --dry-run-wrapper
```

 Which will output something along the lines of

Dry-run result: Transaction is valid. Gas used: 1785; This means that we could reasonably make this transfer transaction with a gas-limit of 2000.

Hence, when making the transfer, we could specify the gas-limit as follows:

```
namadac
```

```
transfer \ --source
```

```
my-new-acc \ --target
```

```
validator-1 \ --token
```

```
NAM \ --amount
```

```
10 \ --signing-keys
```

```
keysha \ --gas-payer
```

```
accountant \ --gas-limit
```

```
2000
```

 If for some reason, we wanted to pay a higher gas fee, we could also specify that as follows:

```
namadac
```

```
transfer \ --source
```

```
my-new-acc \ --target
```

```
validator-1 \ --token
```

```
NAM \ --amount
```

```
10 \ --signing-keys
```

```
keysha \ --gas-payer
```

```
accountant \ --gas-limit
```

```
2000 \ --gas-price
```

```
0.01
```

 This will incentivise validators to prioritise this transaction above those with a lower gas price.

## Paying fees with tokens in the MASP

It is also possible to pay for fees using the MASP. The purpose of this is to ensure that even though a user may not have NAM in their transparent balance, they can still make transparent transactions on chain. This is yet another incentive for users to keep the maximum amount of assets in the MASP.

In order to pay for fees using the MASP, the user must specify the `--gas-spending-key` flag, and set it the alias of a spending key in your wallet. This will mean that the fees will be deducted from the shielded balance of the spending key, and unshielded to the transparent balance of the `--gas-payer` (or the address corresponding to the first key in the `--signing-keys`), before being paid for by the `--gas-payer`.

For example, if the user has a spending key `spending-key-1` in their wallet, and they want to pay for the fees of a transaction

using the MASP, they would run the following command:

```
namadac
```

```
transfer \ --source
```

```
keysha \ --target
```

```
address-b \ --token
```

```
OSMO \ --amount
```

```
10 \ --gas-payer
```

```
keysha \ --gas-spending-key
```

spending-key-1 In this example, keysha may only have an OSMO balance in their transparent balance, but spending-key-1 may have a positive NAM balance in their shielded balance. In this case, the NAM will be unshielded to the transparent balance of keysha, and then used to pay for the transaction fee.

## Using a disposable gas payer

It is also possible to use a disposable gas payer to pay for transaction fees. This is useful in cases where the user does not want to reveal the identity of the --gas-payer. In order to use a disposable gas payer, the user must include the --disposable-gas-payer flag, AND also must specify a --gas-spending-key. This will mean that the fees will be deducted from the shielded balance of the --gas-spending-key, and unshielded to the transparent balance of an ephemeral transparent address before being paid for by the ephemeral address.

For example, if the user has a spending key spending-key-1 in their wallet, and they want to pay for the fees of a transaction using a disposable address, they would run the following command:

```
namadac
```

```
transfer \ --source
```

```
keysha \ --target
```

```
address-b \ --token
```

```
OSMO \ --amount
```

```
10 \ --gas-spending-key
```

```
spending-key-1 \ --disposable-gas-payer
```

[Shielded Transfers Bonding and Proof-of-stake](#)