



## ZooBC CLI Manual

zoobc-cli — one binary, every transaction

[Theme](#)

One command-line tool that builds, signs, and submits **every** ZooBC transaction — 50 commands from a plain `send-zbc` to multisig, node registration, the exchange, on-chain storage, and governance. Machine-friendly by default (JSON in, JSON out) with an interactive mode for humans.

### 01 Get it

Download the Linux x86\_64 build, unpack, and run. No install, no dependencies.

[↓ Download zoobc-cli \(Linux x86\\_64\)](#)

```
# unpack and put it on your PATH
tar xzf zoobc-cli-linux-x86_64.tar.gz
sudo mv zoobc-cli /usr/local/bin/

# point it at any node or gateway (default: http://localhost:8080)
zoobc-cli list --api https://node.proofofparticipation.network
```

Other platforms are coming; for now the build is Ubuntu / Linux x86\_64.

### 02 Three ways to run it

#### Default — JSON in, JSON out

Pass params as positional arguments; get a JSON result back. This is the mode other apps and scripts call. The **first parameter is always the sender private key**.

```
zoobc-cli send-zbc <privkey> ZBC_4HFG43GR_... 1000000000 --api $API  
→ {"transaction_hash":"a5f1...", "amount":1000000000, ...}
```

## Interactive — --verbose

Prompts you for each field one at a time, then prints a human-readable result. Great for one-offs.

```
zoobc-cli issue-token --verbose --api $API  
Sender private key: ... · Symbol: ... · Name: ... · Decimals: ...
```

## Piped JSON — --json-input

Read the parameters as a JSON object on stdin — the keys are exactly what `help <cmd> lists`.

```
echo '{"sender_privkey":"...", "recipient":"ZBC_...", "amount":"1000000000"}' \  
| zoobc-cli send-zbc --json-input --api $API
```

## 03 Discover what to send

Two built-ins tell you everything — no need to read source.

```
zoobc-cli list          # every command, grouped by category  
zoobc-cli help multisig # the JSON fields for ONE command
```

`help <cmd>` prints each field's key, whether it's required, its default, and a ready-to-edit sample JSON object:

```
multisig — N-of-M multisig SendZBC (inner tx + participant signatures) (tx typ  
  sender_privkey    (required) Sender private key (64 hex)  
  participants      (required) comma-separated participant ZBC addresses  
  min_signatures    (required) required signatures (N of M)  
  nonce             (optional) multisig account nonce [default: 0]  
  signer_privkeys   (required) participant keys that sign now
```

|           |                                                           |
|-----------|-----------------------------------------------------------|
| recipient | (required) inner SendZBC recipient address                |
| amount    | (required) inner SendZBC amount (atomic units)            |
| inner_fee | (optional) inner SendZBC fee (atomic) [default: 10000000] |

Sample: {"sender\_privkey":"...", "participants":"...", ...}

## 04 Common flags & conventions

|                                |                                                                                                                                                                 |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>--api &lt;url&gt;</code> | Node or gateway endpoint. Default <code>http://localhost:8080</code> . Point it at <code>https://node.proofofparticipation.network</code> for the live network. |
| <code>--fee &lt;n&gt;</code>   | Override the transaction fee (atomic ZBC). Omit to use the default / size-aware minimum.                                                                        |
| <code>--verbose</code>         | Interactive prompts + human-readable output (see mode 2).                                                                                                       |
| <code>--json-input</code>      | Read params as a JSON object on stdin (mode 3).                                                                                                                 |

All amounts and prices are **atomic integers** (ZBC  $\times 1e8$ ; token amounts  $\times 10^{\text{decimals}}$ ; order prices  $\times 1e8$ ). Addresses may be a full ZBC\_... string or hex. The tool never crashes on bad input — it returns a JSON error object (or a text error in `--verbose`).

## 05 Worked examples

```
# transfer a colored token
zoobc-cli transfer-token $KEY ZBC_6FINNB6J_... <token_id> 500 --api $API

# register a validator node (owner signs; node key supplies the pubkey; PoOwner
zoobc-cli register-node $OWNER_KEY $NODE_KEY 1000000000000 --api $API

# open an N-of-M multisig and sign with your key (others co-sign later)
zoobc-cli multisig $KEY "ZBC_A...,ZBC_B...,ZBC_C..." 2 0 "$KEY" ZBC_A... 1000000000 5000

# schedule a vesting/recurring payment: 1 ZBC per fire, 12 fires, monthly
zoobc-cli scheduled-transfer $KEY ZBC_... 0 1000000000 2592000 12 0 0 0 0 --api $A
```

## 06 All 50 commands

Byte-exact body layouts for every transaction are in the [Transaction Manual](#). Run `zoobc-cli help <cmd>` for a command's fields.

## VALUE & TOKENS

|                     |                                             |
|---------------------|---------------------------------------------|
| send-zbc            | Send ZBC to an address                      |
| liquid-payment      | Stream ZBC over time (vesting)              |
| liquid-payment-stop | Stop a liquid payment (settle pro-rata)     |
| transfer-token      | Transfer a held colored token               |
| issue-token         | Issue a colored-coin token                  |
| mint-token          | Mint more of a mintable token               |
| burn-token          | Burn a token (redeem backing if redeemable) |
| finance-token       | Top up a token's survival financing         |

## EXCHANGE & MARKETPLACE

|               |                                       |
|---------------|---------------------------------------|
| market-create | Open a (base,quote) order-book market |
| order-place   | Place a limit/market order            |
| order-cancel  | Cancel a resting order                |
| swap-create   | Create an atomic swap offer           |
| swap-accept   | Accept (fill) a swap offer            |
| swap-cancel   | Cancel an open swap offer             |

## APPS

|            |                                              |
|------------|----------------------------------------------|
| app-create | Create an app (2-player or solo-vs-house)    |
| app-join   | Join an open app                             |
| app-move   | Submit a move                                |
| app-resign | Resign an app                                |
| app-claim  | Claim a timed-out app                        |
| app-settle | Settle a state-channel app (replay vouchers) |

## STORAGE & DATA

|                     |                                                       |
|---------------------|-------------------------------------------------------|
| store-file          | Anchor a decentralized-storage file (manifest + rent) |
| add-prepaid-storage | Fund an account's storage rent                        |

|                                                                   |                                                                  |
|-------------------------------------------------------------------|------------------------------------------------------------------|
| setup-dataset                                                     | Set an account-dataset property→value                            |
| remove-dataset                                                    | Remove an account-dataset property                               |
| delete-dataset                                                    | Delete a dataset object (refunds its deposit)                    |
| set-dataset-policy                                                | Set a dataset object's manage policy                             |
| transfer-dataset                                                  | Propose transfer of a dataset object                             |
| accept-dataset                                                    | Accept a pending dataset transfer                                |
| <b>ACCOUNT: ESCROW · SCHEDULER · TRIGGERS · MULTISIG · VOTING</b> |                                                                  |
| multisig                                                          | N-of-M multisig SendZBC (inner tx + signatures)                  |
| escrow-request                                                    | Create a recipient-initiated escrow request                      |
| approve-escrow                                                    | Approve / reject / expire an escrow                              |
| scheduled-transfer                                                | Vesting / recurring transfers on a timer                         |
| cancel-schedule                                                   | Cancel a pending scheduled transfer                              |
| reassign-schedule                                                 | Redirect a schedule to a new recipient                           |
| create-trigger                                                    | Schedule a future SendZBC (keeper/oracle)                        |
| cancel-trigger                                                    | Cancel a pending trigger (refund)                                |
| attest-event                                                      | Attest an external event (oracle node)                           |
| fee-vote-commit                                                   | Commit phase of the fee-scale vote                               |
| fee-vote-reveal                                                   | Reveal phase of the fee-scale vote                               |
| <b>NODE OPERATOR</b>                                              |                                                                  |
| register-node                                                     | Register a validator node (owner signs; builds ProofOfOwnership) |
| update-node                                                       | Update a node's locked balance / registration                    |
| remove-node                                                       | Remove a node from the validator set                             |
| claim-node                                                        | Claim a previously-removed registration                          |
| <b>GATEWAY OPERATOR</b>                                           |                                                                  |
| register-gateway                                                  | Register a gateway (key + domain + url, locks stake)             |
| unregister-gateway                                                | Withdraw a gateway; refund the stake                             |

|                                   |                                                                                   |
|-----------------------------------|-----------------------------------------------------------------------------------|
| <a href="#">gateway-heartbeat</a> | Prove a gateway is alive — signed by the GATEWAY key, but any sender may relay it |
|-----------------------------------|-----------------------------------------------------------------------------------|

---

## GOVERNANCE (RELEASE AUTHORITY)

|                                           |                                     |
|-------------------------------------------|-------------------------------------|
| <a href="#">register-release</a>          | Publish a signed binary release     |
| <a href="#">revoke-release</a>            | Revoke a published release          |
| <a href="#">release-authority-propose</a> | Propose a new release authority     |
| <a href="#">release-authority-accept</a>  | Accept a release-authority handover |

---

Part of the [ZooBC Manuals](#). See the [Transaction Manual](#) for byte layouts · open the [wallet](#) or [explorer](#) · back to [the gateway home](#).