TRP VestingEscrow

- Source Code
- Deployed Contracts (mainnet)* VestingEscrowFactory
-
    - VestingEscrowProto
-
    - VotingAdapter
- Deployed Contracts (goerli+prater)* VestingEscrowFactory
-
    - VestingEscrowProto
-
    - VotingAdapter
- Detailed contracts spec

Token Reward Program (TRP) escrow contracts allow transparent on-chain distribution and vesting of the token rewards for the Lido DAO contributors.

VestingEscrowFactory

Public variables

- voting_adapter: address
-
    - address of the VotingAdapter used in the vestings
- owner: address
-
    - factory and vestings owner
- manager: address
-
    - vestings manager

View methods

target()

Returns immutableTARGET

@external @view def target() -> uint256

token()

Returns immutableTOKEN

@external @view def token() -> uint256

Methods

deploy_vesting_contract()

note Before callingdeploy_vesting_contract() caller need to have enough tokens on the balance and callapprove(vestingFactoryAddress, fundAmount) on the token contract Deploy and fund a new instance of theVestingEscrow for the givenrecipient . Set all params for the deployed escrow. Returns address of the deployed escrow

@external def deploy_vesting_contract(amount: uint256, recipient: address, vesting_duration: uint256, vesting_start: uint256 = block.timestamp, cliff_length: uint256 = 0, is_fully_revokable: bool = False) -> address

Parameters

Name Type Description amount uint256 Amount of the tokens to be controlled by vesting recipient address Recipient of the vested funds vesting_duration uint256 Vesting duration in seconds vesting_start uint256 Vesting start time in seconds (unix time in sec) cliff_length uint256 Cliff duration in seconds is_fully_revokable bool Flag that enablesrevoke_all method note Reverts if any of the following is true:

- vesting_duration <= 0
- .

- cliff_length >= vesting_duration
- token transfer from caller to factory fails
- approve of the tokens to the actual vesting fails

recover_erc20()

Collect ERC20 tokens from the contract to theowner .

@external def recover_erc20(token: address, amount: uint256)

Parameters

Name Type Description token address Address of ERC20 token to recover amount uint256 Amount of the tokens to recover note Reverts if:

- tokens transfer toowner
- fails

recover_ether()

Collect all ether from the contract to theowner .

@external def recover_ether() note Reverts if:

- Ether transfer toowner
- fails

update_voting_adapter()

Setself.voting_adapter tovoting_adapter .

@external def update_voting_adapter(voting_adapter: address)

Parameters

Name Type Description voting_adapter address New voting adapter note Reverts if:

- called by anyone exceptVestingEscrowFactory
- owner

change_owner()

Setself.owner toowner .

@external def change_owner(owner: address)

Parameters

Name Type Description owner address Newowner address note Reverts if:

- called by anyone exceptVestingEscrowFactory
- owner
- argowner
- is empty address

change_manager()

Setself.manager tomanager .

@external def change_manager(manager: address)

Parameters

Name Type Description manager address Newmanager address note Reverts if:

- called by anyone exceptVestingEscrowFactory
- owner

VestingEscrow

Public variables

- recipient: address
  - address that can claim tokens from escrow
- token: ERC20
  - address of the vested token
- start_time: uint256
  - vesting start time (UTC time in UNIX seconds)
- end_time: uint256
  - vesting end time (UTC time in UNIX seconds)
- cliff_length: uint256
  - cliff length in seconds
- factory: IVestingEscrowFactory
  - address of the parent factory
- total_locked: uint256
  - total amount of the tokens to be vested (does not change after claims)
- is_fully_revokable: bool
  - flag showing if the escrow is fully revocable or not
- total_claimed: uint256
  - total amount of the claimed tokens
- disabled_at: uint256
  - effective vesting end time (UTC time in UNIX seconds). Can differ from end_time in case of the revoke_xxx methods call
- initialized: bool
  - flag indicating that escrow was initialized
- is_fully_revoked: bool
  - flag indicating that escrow was fully revoked and there are no more tokens

View methods

unclaimed()

Returns the current amount of the tokens available for the claim.

@external @view def unclaimed() -> uint256

locked()

Returns the current amount of the tokens locked.

@external @view def locked() -> uint256

Methods

claim()

Claim tokens to thebeneficiary address. If the requested amount is larger thanunclaimed , then theunclaimed amount will be claimed.

Returns actual amount of the tokens claimed.

@external def claim(beneficiary: address = msg.sender, amount: uint256 = max_value(uint256))

Parameters

Name Type Description beneficiary address Address to claim tokens to amount uint256 Amount of the tokens to claim note Reverts if:

- called by anyone except vestingrecipient
- tokens transfer tobeneficiary
- fails

revoke_unvested()

Disable further flow of tokens and revoke the unvested part to the owner.

@external def revoke_unvested() note Reverts if:

- called by anyone exceptVestingEscrowFactory
- owner or manager
- tokens transfer toVestingEscrowFactory.owner()
- fails

revoke_all()

Disable further flow of tokens and revoke all tokens to the owner.

@external def revoke_all() note Reverts if:

- is_fully_revocable
- param of theVestingEscrow
- is not True
- called by anyone exceptVestingEscrowFactory
- owner
- tokens transfer toVestingEscrowFactory.owner
- fails

recover_erc20()

Collect ERC20 tokens from the contract to therecipient .

@external def recover_erc20(token: address, amount: uint256)

Parameters

Name Type Description token address Address of ERC20 token to recover amount uint256 Amount of the tokens to recover note Reverts if:

- tokens transfer torecipient
- fails

recover_ether()

Collect all ether from the contract to therecipient .

@external def recover_ether() note Reverts if:

- Ether transfer torecipient
- fails

aragon_vote()

Participate in the Aragon vote using all available tokens on the contract's balance. UsesdelegateCall toVotingAdapter .VotingAdapter address is fetched fromself.factory .

@external def aragon_vote(abi_encoded_params: Bytes[1000])

Parameters

Name Type Description abi_encoded_params Bytes[1000] ABI encoded params for thevote method call. can be compiled usingVotingAdapter.encode_aragon_vote_calldata note Reverts if:

- called by anyone except vestingrecipient

snapshot_set_delegate()

Delegate Snapshot voting power of all available tokens on the contract's balance todelegate . UsesdelegateCall toVotingAdapter .VotingAdapter address is fetched fromself.factory .

@external def snapshot_set_delegate(abi_encoded_params: Bytes[1000])

Parameters

Name Type Description abi_encoded_params Bytes[1000] ABI encoded params for thedelegate method call. can be compiled usingVotingAdapter.encode_snapshot_set_delegate_calldata note Reverts if:

- called by anyone except vestingrecipient

delegate()

note Stub at the moment of writing Delegate voting power of all available tokens on the contract's balance todelegate . UsesdelegateCall to VotingAdapter.VotingAdapter address is fetched fromself.factory .

@external def delegate(abi_encoded_params: Bytes[1000])

Parameters

Name Type Description abi_encoded_params Bytes[1000] ABI encoded params for thevote method call. can be compiled usingVotingAdapter.encode_delegate_calldata note Reverts if:

- called by anyone except vestingrecipient

VotingAdapter

Public variables

- owner: address
-
    - votingAdapter owner

View methods

encode_aragon_vote_calldata()

Returns abi encoded params for thearagon_vote call.

@external @view def encode_aragon_vote_calldata(voteId: uint256, supports: bool) -> Bytes[1000]

Parameters

Name Type Description voteId uint256 Aragon vote id amount bool Supports flag.True - for,False - against

encode_snapshot_set_delegate_calldata()

Returns abi encoded params for thesnapshot_set_delegate call.

@external @view def encode_snapshot_set_delegate_calldata(delegate: address) -> Bytes[1000]

Parameters

Name Type Description delegate address Address to delegate snapshot voting power to

encode_delegate_calldata()

Returns abi encoded params for thedelegate call.

@external @view def encode_delegate_calldata(delegate: address) -> Bytes[1000]

Parameters

Name Type Description delegate address Address to delegate voting power to

Methods

aragon_vote()

Participate in the Aragon vote using all available tokens on the contract's balance. It makes sense only for delegateCalls, so the caller's balance will be used. UsesVOTING_CONTRACT_ADDR as the voting contract address.

@external def aragon_vote(abi_encoded_params: Bytes[1000])

Parameters

Name Type Description abi_encoded_params Bytes[1000] ABI encoded params for thevote method call. can be compiled usingVotingAdapter.encode_aragon_vote_calldata note Reverts if:

- called by anyone except vestingrecipient

snapshot_set_delegate()

Delegate Snapshot voting power of all available tokens. Makes sense only for delegateCalls so that the balance of the caller will be used. UsesSNAPSHOT_DELEGATE_CONTRACT_ADDR as the voting contract address.

@external def snapshot_set_delegate(abi_encoded_params: Bytes[1000])

Parameters

Name Type Description abi_encoded_params Bytes[1000] ABI encoded params for thedelegate method call. can be compiled usingVotingAdapter.encode_snapshot_set_delegate_calldata note Reverts if:

- called by anyone except vestingrecipient

delegate()

note Stub at the moment of writing Stub for the future implementation of the Voting with Delegation.

@external def delegate(abi_encoded_params: Bytes[1000])

Parameters

Name Type Description abi_encoded_params Bytes[1000] ABI encoded params for thevote method call. can be compiled usingVotingAdapter.encode_delegate_calldata note Always reverts

recover_erc20()

Collect ERC20 tokens from the contract to theowner .

@external def recover_erc20(token: address, amount: uint256)

Parameters

Name Type Description token address Address of ERC20 token to recover amount uint256 Amount of the tokens to recover note Reverts if:

- tokens transfer toowner
- fails

recover_ether()

Collect all ether from the contract to theowner .

@external def recover_ether() note Reverts if:

- Ether transfer toowner
- fails

change_owner()

Setself.owner toowner

@external def change_owner(owner: address)

Parameters

Name Type Description owner address Newowner address note Reverts if:

- called by anyone exceptVotingAdapter
- owner
- argowner
- is empty address Edit this page Previous MevBoostRelayAllowedList Next GateSeal