Chief - Detailed Documentation

Electing a Chief contract via an approval voting system * Contract Name: * chief.sol * Type/Category: * Governance Module * Associated MCD System Diagram * Contract Source * Etherscan *

1. Introduction (Summary)

TheDs-Chief smart contract provides a method to elect a "chief" contract via an approval voting system. This may be combined with another contract, such asDSAuthority , to elect a ruleset for a smart contract system.

In short, voters will lock up their voting tokens in order to give their votes some weight in the system. The voting is then done by continuousapproval voting , where users receive IOU tokens when they lock their voting tokens up, which is useful for secondary governance mechanisms. The IOU tokens may not be exchanged for the locked tokens except by individuals who have actually locked funds in the contract itself, and only up to the amount they have locked.

?

1. Contract Details

Glossary (Chief)

DSChiefApprovals provides the following public properties:

- slates
- : A mapping ofbytes32
- toaddress
- arrays. Represents sets of candidates. Weighted votes are given to slates.
- votes
- : A mapping of voter addresses to the slate they have voted for.
- approvals
- : A mapping of candidate addresses to theiruint
- weight.
- deposits
- : A mapping of voter addresses touint
- number of tokens locked.
- GOV
- :DSToken
- used for voting.
- IOU
- :DSToken
- issued in exchange for lockingGOV
- tokens.
- hat
- : Contains the address of the current "chief."
- MAX_YAYS
- : Maximum number of candidates a slate can hold.
-

Most of the functions are decorated with the thenote modifier fromds-note , meaning that they fire a standardized event when called. Additionally, one custom event is also provided:

- Etch(bytes32 indexed slate)
- : Fired when a slate is created.
-

1. Key Mechanisms & Concepts

There are two contracts in ds-chief:

1. DSChiefApprovals
2. DSChief, which inherits from DSChiefApprovals.
3.

Key Functionalities (as defined in the smart contract)

DSChiefApprovals

DSChiefApprovals(DSToken GOV_, DSToken IOU_, uint MAX_YAYS_)

- The constructor. SetsGOV
- ,IOU
- , andMAX_YAYS
- .
-

lock(uint wad)

- Charges the userwad
- GOV
- tokens, issues an equal amount ofIOU
- tokens to the user, and addswad
- weight to the candidates on the user's selected slate. Fires aLogLock
- event.
-

free(uint wad)

- Charges the userwad
- IOU
- tokens, issues an equal amount ofGOV
- tokens to the user, and subtractswad
- weight from the candidates on the user's selected slate. Fires aLogFree
- event.
-

etch(address[] yays) returns (bytes32 slate)

- Save a set of ordered addresses as aslate
- and return a unique identifier for it.
-

vote(address[] yays) returns (bytes32 slate)

- Save a set of ordered addresses as aslate
- , moves the voter's weight from their current slate to the new slate, and returns the slate's identifier.
-

vote(bytes32 slate)

- Removes voter's weight from their current slate and adds it to the specified slate.
-

lift(address whom)

- Checks the given address and promotes it tos/chief/hat
- if it has more weight than the currents/chief/hat
- .
-

DSChief

DSChief is a combination ofDSRoles from theds-roles package andDSChiefApprovals . It can be used in conjunction withds-auth (as an authority object) to govern smart contract systems.

Public Functions

DSChief(DSToken GOV_, DSToken IOU_, uint MAX_YAYS_)

- The constructor. SetsGOV
- ,IOU
- , andMAX_YAYS
- .
-

setOwner(address owner_)

- Reverts the transaction. Overridden fromDSAuth
- .

-

setAuthority(DSAuthority authority_)

- Reverts the transaction. Overridden fromDSAuth
- .
-

isUserRoot(address who) constant returns (bool)

- Returnstrue
- if the given address is the chief.
-

setRootUser(address who, bool enabled)

- Reverts the transaction. Overridden fromDSRoles
- .
-

DSRoles

Seeds-roles for inherited features.

1. Gotchas (Potential source of user error)

In general, when we refer to the"hat" , it can be any address — be it a single-use contract like ds-spell, a multi-use contract or an individual's wallet. Thus, ds-chief can work well as a method for selecting code for execution just as well as it can for realizing political processes.

Example:

Theds-chief could be used as a token-weighted voting system governing another set of smart contracts that uses theds-auth withds-roles . In a scenario such as this, "candidates" would consist of contracts changing the state of the smart contract set under governance. Such a contract being elected as ”hat" would be granted all of the permissions to execute whatever changes are necessary. Theds-chief could also be used within such a contract set in conjunction with a proxy contract, such asds-proxy or a name resolution system like ENS for the purpose of voting in new versions of contracts.

Understanding the IOU Token

The purpose of the IOU token is to allow for the chaining of governance contracts. In other words, this allows you to have a number ofDSChief or other similar contracts use the same governance token by means of accepting the IOU token of theDSChief contract before it is a governance token.

Example:

Let’s say there are threeDSChief contracts (chiefA, chiefB, and chiefC) and achiefA.GOV that is the MKR token. If we setchiefB.GOV tochiefA.IOU andchiefC.GOV tochiefB.IOU , this allows all three contracts to run using a common group of MKR.

Approval Voting

This type of voting is when each voter selects which candidates they approve of, with the top n "most approved" candidates being then elected. Each voter can cast up to n + k votes, where k equals some non-zero positive integer. This way voters to move their approval from one candidate to another without needing to first withdraw support from the candidate being replaced. Without this in place, moving approval to a new candidate could result in a less-approved candidate moving momentarily into the set of the elected candidates.Note: In the case ofds-chief , n is equal to 1.

In addition, theds-chief weighs votes according to the quantity of a voting token that has been chosen to lock up in theDSChief or theDSChiefApprovals contract. It's important to note that the voting token used in ads-chief deployment must be specified at the time of deployment and cannot be changed afterward.

Implementations

If you are writing a front-end UI for this smart contract, please note that the address[] parameters that are passed to theetch andvote functions must be byte-ordered sets.

Example:

Using[0x0, 0x1, 0x2, ...] is valid but using[0x1, 0x0, ...] and[0x0, 0x0, 0x1, ...] is not. This ordering constraint allows the

contract to cheaply ensure voters cannot multiply their weights by listing the same candidate on their slate multiple times.

1. Failure Modes (Bounds on Operating Conditions & External Risk Factors)
2. MKR users moving their votes from one spell to another
3.
    - One of the biggest potential failure modes occurs when people are moving their votes from one spell to another. This opens up a gap/period of time when only a small amount of MKR is needed to lift a random hat.
4. *
5. Lift is not called on spells that have more MKR than the current hat
6.
    - The only way a spell can get the hat is iflift
7.
    - is called on it. So, even if a spell gains much more MKR on it than the hat, iflift
8.
    - is never called on it, the hat will remain on a spell that no longer has the most MKR. This could lower the bar for the amount of MKR needed to pass something, potentially making the system less safe.
9. *
10. Stray spells without expiration
11.
    - Due to the continuous nature of voting, a spell will remain live in the system even if it was not approved to be the governing proposal. This means that MKR holders can continue to vote on the candidate and in times of lower voter participation there is potential for them to introduce a failure mode by voting for an unexpected and/or older candidate. This illustrates why increased voter participation is important and that a higher amount of MKR on the current governing proposal adds to the stability of the system.
12. *
13. Unsafe states when migrating to a new chief contract
14.
    - When migrating to a new chief, authority must be transferred to the new contract and revoked from the old. This poses a small coordination problem as the new contract must already have enough MKR on itshat
15.
    - to be safe against governance attacks while the voters enacting the change itself must have enough MKR in the old chief to pass the proposal.
16. *
17.

Previous Pause - Detailed Documentation Next Rates Module Last updated2 years ago On this page *

Export as PDF