Spell - Detailed Documentation

- Contract Name:
- spell.sol
- Type/Category:
- Governance Module
- Associated MCD System Diagram
- Contract Source
-

1. Introduction (Summary)

ADSSpell is an un-owned object that performs one action or series of atomic actions (multiple transactions) one time only. This can be thought of as a one-off DSProxy with no owner (no DSAuth mix-in, it is not a DSThing).

This primitive is useful to express objects that do actions which shouldn't depend on "sender", like an upgrade to a contract system that needs to be given root permission. By convention, it is usually what is used to change SCD or MCD system parameters (where it is given auth via voting inds-chief).

?

1. Contract Details

Thespell.sol contract contains two main contracts: DSSPELL and DSSpellBook. DSSPELL is the core contract that, with call instructions set in the constructor, can actually perform the one-time action. DSSpellBook is a factory contract designed to make the creation of DSSPELLs easier.

Glossary (Spell)

- whom
-
  - is the address the spell is targeting, usually SAI_MOM in SCD.
- mana
-
  - is the amount of ETH you are sending, which in spells it is usually 0.
- data
-
  - bytes memory calldata.
- done
-
  - indicates that the spell has been called successfully.
-

1. Key Mechanisms & Concepts
2. hat
3.
   - A spell comes into effect as the hat when someone calls the lift function. This is only possible when the spell in question has more MKR voted towards it than the current hat.
4. cast
5.
   - Once a spell has become the hat, it can be cast and its new variables will go into effect as part of the live Maker system. It is worth noting that a spell can only be cast once.
6. lift
7.
   - The process whereby a new spell replaces the old proposal.
8.

Note: thehat andlift have more to do withds-chief thands-spell but are important to mention here for context.

Immutable Actions

whom ,mana , anddata are set in the constructor, so the action a spell is to perform cannot be changed after the contract has been deployed.

1. Gotchas (Potential source of user error)

Note that the spell is only marked as "done" if the CALL it makes succeeds, meaning it did not end in an exceptional condition and it did not revert. Conversely, contracts that use return values instead of exceptions to signal errors could be successfully called without having the effect you might desire. "Approving" spells to take action on a system after the spell is deployed generally requires the system to use exception-based error handling to avoid griefing.

1. Failure Modes (Bounds on Operating Conditions & External Risk Factors)

2. spell

3.
    - A spell may remain uncast if it did not reach the required amount of MKR in order to pass. If this occurs, the spell may remain available as a later target if enough MKR is voted towards it.
4. lift
5.
    - Although spells cannot be cast a second time, they can be lifted to become the hat more than once if enough MKR votes remain on that proposal. The proposals parameters will not go into effect, however any additional spell will need to have more than that amount of MKR voted towards it in order to become the new hat. Seeforum post
6. for a description of this having once occurred.
7. cast
8.
    - If, whencast
9. is called, the spell's one-time action fails,done
10. does not get flipped and the spell remains castable.
11.

Previous Governance Module Next Pause - Detailed Documentation Last updated4 years ago On this page *1. Introduction (Summary) * 2. Contract Details * Glossary (Spell) * 3. Key Mechanisms & Concepts * 4. Gotchas (Potential source of user error) * 5. Failure Modes (Bounds on Operating Conditions & External Risk Factors)

Export as PDF