

<<Contract>>

<<Fixed>>

Token

NAME : string

DECIMAL : Integer

SYMBOL : string

motd : string

controller : Controller

<<new>> burnAddress : address

<<new>> burnable : bool

<<event>> Motd(message : string)

setMotd(_m : string)

setController(_c : address)

<<const>> balanceOf(a : address) : uint

<<const>> totalSupply() : uint

<<const>> allowance(_owner : address,_spender : address) : uint

<<onlyPayloadSize, notPaused>> transfer(_to : address,_value : address) : bool

<<onlyPayloadSize, notPaused>> transferFrom(_from : address,_to : address,_value : uint) : bool

<<onlyPayloadSize, notPaused>> approve(_spender : address,_value : uint) : bool

<<onlyPayloadSize, notPaused>> increaseApproval(_spender : address,_addedValue : uint) : bool

<<onlyPayloadSize, notPaused>> decreaseApproval(_spender : address,_subtractedValue : uint) : bool

<<modifier>> onlyPayloadSize(numwords : uint)

<<notPaused>> burn(_amount : uint)

<<modifier>> onlyController()

<<onlyController>> controllerTransfer(_from : address,_to : address,_value : uint)

<<onlyController>> controllerApprove(_owner : address,_spender : address,_value : uint)

<<new, onlyController>> controllerBurn(_from : address,_to : address,_value : uint)

<<new, onlyController>> setBurnAddress(_address : address)

<<new, onlyController>> enableBurning()

<<new, onlyController>> disableBurning()

<<new, modifier>> burnEnabled()

<<new, notPaused, burnEnabled>> burn(_to : bytes32,_amount : uint) : bool

<<new, notPaused, burnEnabled>> claimByProof(data : bytes32[],proofs : bytes32[],number : uint256) : bool

<<new, notPaused, burnEnabled>> claim() : bool

<<new, onlyController>> controllerBurnClaim(_claimer : address,_value : uint256)