7245

CONTRACT ID: 0x6816dF892602c7E6b741f03FEf9c7eC57061A73f

Transaction

0x980caab45bd3478f9e7c307d398941d998f32636944cd687081d2747a0ca2c97

TxHash:
0x980caab45bd3478f9e7c307d398941d998f32636944cd687081d2747a0ca2c97

TxReceipt Status:
Success

Block Height:
5083867 (4 block confirmations)

TimeStamp:
1 min ago (Feb-13-2018 04:43:35 PM +UTC)

From:
0x515581325a538a3dd7fccbe3a1cd6cf054f52a62

To:
[Contract 0x6816df892602c7e6b741f03fef9c7ec57061a73f Created]

Value:
0 Ether ($0.00)

Gas Limit:
1000000

Gas Used By Txn:
681565

Gas Price:
0.000000028 Ether (28 Gwei)

Actual Tx Cost/Fee:
0.01908382 Ether ($16.03)

Cumulative Gas Used:
7206853

Nonce:
14

Input Data:

```
0x60a060405260046060527f48302e3100000000000000000000000000000000000000000000000000
00000060805260068054600082905 27f48302e310000000000000000000000000000000000000000
0000000000000000882556100b5907ff652222313e28459528d920b65115c16c04f3efc82aaedc97be59f
3f377c0d3f602060026001841615610100026000190190931692909204601f0191909104810190 5b80
82111561017957600081556001016100a1565b50506040516109 4b38038061094b83398101604052 8
080519060200190919080518201919060200180519060200190919080518201919060200150508360
0060005060003360 0160a060020a03168152602001908152602001600020600050819055508360026
00050819055508260036000509080519060200190828054600181600116156101000203166002900 4
906000526020600020906 01f016020900481019282601f1061017d57805160ff1916838001178555 5b5
06101ad9291506100a1565b5090565b828001600101855558215610 16d579182015b828111156101 6d
5782518260005055916020019190600101906101 8f565b50506004805460ff19168317905560058054
825160008390527f036b6384b5eca791c62761152d0c79bb0604c104a5fb6f4eb0703f3154bb3db0602
060026001851615610100026000190190941693909304601f908101849004820193860190839010 61
022d57805160ff1916838001178555 5b5061025d9291506100a1565b828001600101855558215610221
579182015b8281111561022157825182600050559160200191906001019061023f565b50505050505
06106da806102716000396000f3606060405236156100 8d5760e060020a600035046306fdde0381146
10095578063095ea7b3146100f357806318160ddd1461016857806323b872dd14610171578063313c
e5671461025c57806354fd4d501461026857806370a08231146102c657806395d89b41146102f45780
63a9059cbb14610352578063cae9ca51146103f7578063dd62ed3e146105be575b6105f2610002565b
604080516003805460206002600183161561010002600019019092169190910 4601f8101829004820
284018201909452838352610 5f493908301828280156106b75780601f1061068c5761010080835404 0
283529160200191 6106b7565b61066260043560243533600160a060020a039081166000818152600 1
602090815260408083209487168084529482528083208690558051868152905192949392 7f8c5be1e
5ebec7d5bd14f71427d1e84f3dd0314c0f7b2291e5b200ac8c7c3b925929181900390910190a3506001
5b92915050565b6102e26002548156 5b6106626004356024356044356001 60a060020a0383166000 9
08152602081905260408120548290108015906101c457506001602090815260408083203360 0160a0
60020a031684529091528120548290101 55b80156101d05750600082115b156106bf57600160a0600
20a03838116600081815260208181526040808320805488019055888516808452818420805489900 3
9055600183528184203390961684529482529182902080548790039055815186815291519293927fd
df252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef9281900390910190a3506
0016106c3565b610676600454 60ff1681565b604080516006805460206002600183161561010002600
0190190921691909104601f81018290048202840182019094528383526105f4939083018282801561 0
6b75780601f1061068c5761010080835404028352916020019161 06b7565b600160a060020a036004
3516600090815260208190526040902054 5b60408051918252519081900360200190f35b6105f46005
8054604080516020600260018516156101000260001901909416939093046 01f81018490048402820
184019092528181529291830182828015610 6b75780601f1061068c57610100808354040283529160
2001916106b7565b61066260043560243533600160a060020a03166000908152602081905260408 12
05482901080159061038457506000821 15b156106ca5733600160a060020a03908116600081815260
20818152604080832080548890039055938716808352918490208054870190558351868152935191 9
37fddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef92908190039091019 0
a3506001610162565b60408051602060443560048181013560 1f81018490048402850184019095528
4845261066294813594602480359593946064949293910191819084018382808284375094965050 50
5050505033600160a060020a0390811660008181526001602090815260408083209488168084529 48
25280832087905580518781529051929493927f8c5be1e5ebec7d5bd14f71427d1e84f3dd0314c0f7b2
291e5b200ac8c7c3b92592918190039091019 0a383600160a060020a0316604051808 07f7265636569
7665417070726f76616c28616464726573732c75696e743235362c8152602001 7f616464726573732c
```

```
6279746573290000000000000000000000000000000000000815260200150602e01905060405180910
3902060e060020a900433853086604051856 0e060020a0281526004018085600160a060020a031681
5260200184815260200183600160a060020a031681526020018280519060200190808383829060006
004602084601f0104600f02600301f1509050908101906 01f16801561059657808203805160018360 20
036101000a031916815260200191505b5094505050505050600060405180830381600087 6161da5a03f
19250505015156106d257610002565b6102e26004356024356 00160a060020a03828116600090 8152
6001602090815260408083209385168352929052205461016 2565b005b604051808060200182 81038
2528381815181526020019150805190602001908083838290600060046020846 01f0104600f0260030
1f1509050908101906 01f168015610654578082038051600183602003610100 0a0319168152602001 9
1505b5092505050604051809103 90f35b604080519115158252519081900360200190f35b604080516
0ff909216825251908190036020 0190f35b8201919060005260206000209 05b815481529060010190 6
0200180831161069a57829003601f168201915b50505050508156 5b5060005b939250505056 5b5060
00610162565b5060016106c35600000000000000000000000000000000000000000000152d02c7e14
af68000000000000000000000000000000000000000000000000000000000000000000000008000000000
000000000000000000000000000000000000000000000000000001200000000000000000000000000
0000000000000000000000000000000000000c0000000000000000000000000000000000000000000
000000000000000000000454484f52000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000003544852000000
0000000000000000000000000000000000000000000000000000
```

SMART CONTRACT

```
contract Token {

    /// @return total amount of tokens
    function totalSupply() constant returns (uint256 supply) {}

    /// @param _owner The address from which the balance will be retrieved
    /// @return The balance
    function balanceOf(address _owner) constant returns (uint256 balance) {}

    /// @notice send `_value` token to `_to` from `msg.sender`
    /// @param _to The address of the recipient
    /// @param _value The amount of token to be transferred
    /// @return Whether the transfer was successful or not
    function transfer(address _to, uint256 _value) returns (bool success) {}

    /// @notice send `_value` token to `_to` from `_from` on the condition it is
approved by `_from`
    /// @param _from The address of the sender
    /// @param _to The address of the recipient
    /// @param _value The amount of token to be transferred
    /// @return Whether the transfer was successful or not
    function transferFrom(address _from, address _to, uint256 _value) returns (bool
success) {}
```

```
    /// @notice `msg.sender` approves `_addr` to spend `_value` tokens
    /// @param _spender The address of the account able to transfer the tokens
    /// @param _value The amount of wei to be approved for transfer
    /// @return Whether the approval was successful or not
    function approve(address _spender, uint256 _value) returns (bool success) {}

    /// @param _owner The address of the account owning tokens
    /// @param _spender The address of the account able to transfer the tokens
    /// @return Amount of remaining tokens allowed to spent
    function allowance(address _owner, address _spender) constant returns (uint256
remaining) {}

    event Transfer(address indexed _from, address indexed _to, uint256 _value);
    event Approval(address indexed _owner, address indexed _spender, uint256
_value);
}


/*
This implements ONLY the standard functions and NOTHING else.
For a token like you would want to deploy in something like Mist, see
HumanStandardToken.sol.

If you deploy this, you won't have anything useful.

Implements ERC 20 Token standard: https://github.com/ethereum/EIPs/issues/20
.*/

contract StandardToken is Token {

    function transfer(address _to, uint256 _value) returns (bool success) {
        //Default assumes totalSupply can't be over max (2^256 - 1).
        //If your token leaves out totalSupply and can issue more tokens as time goes
on, you need to check if it doesn't wrap.
        //Replace the if with this one instead.
        //if (balances[msg.sender] >= _value && balances[_to] + _value >
balances[_to]) {
        if (balances[msg.sender] >= _value && _value > 0) {
            balances[msg.sender] -= _value;
            balances[_to] += _value;
            Transfer(msg.sender, _to, _value);
            return true;
        } else { return false; }
    }

    function transferFrom(address _from, address _to, uint256 _value) returns (bool
success) {
        //same as above. Replace this line with the following if you want to protect
against wrapping uints.
        //if (balances[_from] >= _value && allowed[_from][msg.sender] >= _value &&
balances[_to] + _value > balances[_to]) {
        if (balances[_from] >= _value && allowed[_from][msg.sender] >= _value &&
_value > 0) {
            balances[_to] += _value;
            balances[_from] -= _value;
            allowed[_from][msg.sender] -= _value;
            Transfer(_from, _to, _value);
            return true;
        } else { return false; }
    }

    function balanceOf(address _owner) constant returns (uint256 balance) {
        return balances[_owner];
    }

    function approve(address _spender, uint256 _value) returns (bool success) {
        allowed[msg.sender][_spender] = _value;
        Approval(msg.sender, _spender, _value);
        return true;
    }

    function allowance(address _owner, address _spender) constant returns (uint256
remaining) {
      return allowed[_owner][_spender];
    }

    mapping (address => uint256) balances;
    mapping (address => mapping (address => uint256)) allowed;
    uint256 public totalSupply;
}

/*
```

```
This Token Contract implements the standard token functionality
(https://github.com/ethereum/EIPs/issues/20) as well as the following OPTIONAL
extras intended for use by humans.

In other words. This is intended for deployment in something like a Token Factory or
Mist wallet, and then used by humans.
Imagine coins, currencies, shares, voting weight, etc.
Machine-based, rapid creation of many tokens would not necessarily need these
extra features or will be minted in other manners.

1) Initial Finite Supply (upon creation one specifies how much is minted).
2) In the absence of a token registry: Optional Decimal, Symbol & Name.
3) Optional approveAndCall() functionality to notify a contract if an approval() has
occurred.

.*/

contract HumanStandardToken is StandardToken {

   function () {
      //if ether is sent to this address, send it back.
      throw;
   }

   /* Public variables of the token */

   /*
   NOTE:
   The following variables are OPTIONAL vanities. One does not have to include
them.
   They allow one to customise the token contract & in no way influences the core
functionality.
   Some wallets/interfaces might not even bother to look at this information.
   */
   string public name;                   //fancy name: eg Simon Bucks
   uint8 public decimals;                //How many decimals to show. ie. There could
1000 base units with 3 decimals. Meaning 0.980 SBX = 980 base units. It's like
comparing 1 wei to 1 ether.
   string public symbol;                 //An identifier: eg SBX
   string public version = 'H0.1';       //human 0.1 standard. Just an arbitrary
versioning scheme.
```

```
    function HumanStandardToken(
        uint256 _initialAmount,
        string _tokenName,
        uint8 _decimalUnits,
        string _tokenSymbol
        ) {
        balances[msg.sender] = _initialAmount;               // Give the creator all initial
tokens
        totalSupply = _initialAmount;                        // Update total supply
        name = _tokenName;                                   // Set the name for display purposes
        decimals = _decimalUnits;                            // Amount of decimals for display
purposes
        symbol = _tokenSymbol;                               // Set the symbol for display
purposes
    }

    /* Approves and then calls the receiving contract */
    function approveAndCall(address _spender, uint256 _value, bytes _extraData)
returns (bool success) {
        allowed[msg.sender][_spender] = _value;
        Approval(msg.sender, _spender, _value);

        //call the receiveApproval function on the contract you want to be notified. This
crafts the function signature manually so one doesn't have to include a contract in
here just for this.
        //receiveApproval(address _from, uint256 _value, address _tokenContract,
bytes _extraData)
        //it is assumed that when does this that the call *should* succeed, otherwise
one would use vanilla approve instead.

if(!_spender.call(bytes4(bytes32(sha3("receiveApproval(address,uint256,address,byt
es)"))), msg.sender, _value, this, _extraData)) { throw; }
        return true;
    }
}
```

CONTRACT ABI

```
[{"constant":true,"inputs":[],"name":"name","outputs":[{"name":"","type":"string"}],"type":"function"},{"constant":false,"inputs":[{"name":"_spender","type":"address"},{"name":"_value","type":"uint256"}],"name":"approve","outputs":[{"name":"success","type":"bool"}],"type":"function"},{"constant":true,"inputs":[],"name":"totalSupply","outputs":[{"name":"","type":"uint256"}],"type":"function"},{"constant":false,"inputs":[{"name":"_from","type":"address"},{"name":"_to","type":"address"},{"name":"_value","type":"uint256"}],"name":"transferFrom","outputs":[{"name":"success","type":"bool"}],"type":"function"},{"constant":true,"inputs":[],"name":"decimals","outputs":[{"name":"","type":"uint8"}],"type":"function"},{"constant":true,"inputs":[],"name":"version","outputs":[{"name":"","type":"string"}],"type":"function"},{"constant":true,"inputs":[{"name":"_owner","type":"address"}],"name":"balanceOf","outputs":[{"name":"balance","type":"uint256"}],"type":"function"},{"constant":true,"inputs":[],"name":"symbol","outputs":[{"name":"","type":"string"}],"type":"function"},{"constant":false,"inputs":[{"name":"_to","type":"address"},{"name":"_value","type":"uint256"}],"name":"transfer","outputs":[{"name":"success","type":"bool"}],"type":"function"},{"constant":false,"inputs":[{"name":"_spender","type":"address"},{"name":"_value","type":"uint256"},{"name":"_extraData","type":"bytes"}],"name":"approveAndCall","outputs":[{"name":"success","type":"bool"}],"type":"function"},{"constant":true,"inputs":[{"name":"_owner","type":"address"},{"name":"_spender","type":"address"}],"name":"allowance","outputs":[{"name":"remaining","type":"uint256"}],"type":"function"},{"inputs":[{"name":"_initialAmount","type":"uint256"},{"name":"_tokenName","type":"string"},{"name":"_decimalUnits","type":"uint8"},{"name":"_tokenSymbol","type":"string"}],"type":"constructor"},{"anonymous":false,"inputs":[{"indexed":true,"name":"_from","type":"address"},{"indexed":true,"name":"_to","type":"address"},{"indexed":false,"name":"_value","type":"uint256"}],"name":"Transfer","type":"event"},{"anonymous":false,"inputs":[{"indexed":true,"name":"_owner","type":"address"},{"indexed":true,"name":"_spender","type":"address"},{"indexed":false,"name":"_value","type":"uint256"}],"name":"Approval","type":"event"}]
```

READ CONTRACT INFORMATION

1. name   THOR string
2. totalSupply   1000000000000000000000000 uint256
3. decimals   18 uint8
4. version   H0.1 string
5. balanceOf

   _owner (address)

   Query

balance uint256

6. symbol THR string
7. allowance

_owner (address)

,

_spender (address)

Query

remaining uint256