**ERIS ISSUE RECREATION Stack Too Deep Not Displayed**

Description: We have an example where presumably the stack is too deep. The reason I am bringing up this issue is not because of stack depth. Instead, it is related to not receiving any errors upon deployment of the contract.

This error came up in integration of our core identity platform. We eventually had to separate the function into many different functions and it worked, which is another reason we presume it is related to stack depth.

Issue Recreation: Consider the contract below:

contract StackDepth

{

function SetCoreIdentity(bytes32 theUniqueID, bytes32[100] theUniqueIDAttributes,

bytes32 theOwnershipTokenID,bytes32[100] theOwnerIDList, uint[100] theOwnershipStakes,

bytes32 theControlTokenID, uint[100] theControlTokens, bytes32[100] theControlIDList,

bytes32[100] theIdentityRecoveryList, uint theRecoveryCondition,

bool isHumanValue) returns (uint val)

{

val = 5;

}

}

The contract takes many parameters and should return a value of five. After being certain that the input is appropriate, it was verified that the contract did not return anything. The output is null rather than five.

To recreate this, we have an epm and a javascript below.

**EPM**

jobs:

- name: StackDepth

job:

deploy:

contract: StackDepth.sol

wait: true

**JAVASCRIPT FILE**

'use strict'

var contracts = require('eris-contracts')

var fs = require('fs')

var http = require('http')

//Set up addresses for deployed contracts

var address\_CoreIdentity = require('./epm.json').StackDepth

var abi\_COID = JSON.parse(fs.readFileSync('./abi/' + address\_CoreIdentity, 'utf8'))

var accounts = require('./accounts.json')

var chainUrl

var manager

var contract

var server

var hostname = 'localhost';

var port = 8004;

chainUrl = 'http://localhost:1337/rpc'

// Instantiate the contract object manager using the chain URL and the account data.

manager = contracts.newContractManagerDev(chainUrl, accounts.chainbreaker\_root\_000)

// Instantiate the contract object using the ABI and the address.

contract = manager.newContractFactory(abi\_COID).at(address\_CoreIdentity)

//for testing

var theUniqueID = "1111";

var theUniqueIDAttributes = ["20","111","121"];

var theOwnershipTokenID = "1232321";

var theOwnerIDList = ["A1","A2","A3"];

var theOwnershipStakes = [3,3,3];

var theControlTokenID = "32";

var theControlTokens = [5,5,5];

var theControlIDList = ["A1","A2","C1"];

var theIdentityRecoveryList = ["A1","C1",""];

var theRecoveryCondition = 1;

var isHumanValue = new Boolean("false");

//add remaining values to make array size 100

for(var i = theOwnerIDList.length; i < 10; i++)

{

theUniqueIDAttributes[i] = "0";

theOwnerIDList[i] = "0";

theOwnershipStakes[i] = 0;

theControlTokens[i] = 0;

theControlIDList[i] = "0";

theIdentityRecoveryList[i] = "0";

}

contract.SetCoreIdentity(theUniqueID,theUniqueIDAttributes,isHumanValue,

theOwnershipTokenID, theOwnerIDList, theOwnershipStakes,

theControlTokenID, theControlTokens, theControlIDList,

theIdentityRecoveryList, theRecoveryCondition, isHumanValue,

function(error, result)

{

console.log("The result is: " + result);

});