**Ownership Token Testing (These are only public, not helper functions)**

**Testing Environment: Solidity Realtime Compiler**

1. **OwnershipToken(bool isHuman, bytes32 theOwnershipName)**

This is the constructor. Coid can call the function getTokenVal(). If it is 0, no ownership can be made. The input was:

bool isHuman = false;

bytes32 theOwnershipName = “abc123”;

*Passed Test?: YES. (validated by calling getOwnershipName() and getTokenVal()).*

1. **function setOwnershipTokenVals(bytes[100] theOwners, uint[100] theVals)**

*Description:* This function resets everything, and is intended mainly for instantiation. For testing purposes, I changed the size to 4 instead of 100.

*Input:* [“O1”, “O2”, “O3”, “O4”], [5,5,5,10]

*Output:* N/A

*Passed Test?:* YES. (Validated by using functions getOwnersList and getOwnershipVal.)

1. **function getTokenVal() returns (uint val)**

**function setOwnershipID(bytes32 theName)**

**function getOwnershipID() returns (bytes32 theID)**

**function getOwnershipVal(bytes32 addr) returns (uint val)**

These functions are validated by **(1) and (2)**. Also, **setOwnershipID**(“abc131”) is validated by subsequently calling **getOwnershipID.**

1. **function updateOwnershipVal(bytes32 addr, uint val) returns (bool txnPassed)**

This function was validated. Called updateOwnershipVal(“O1”,17), it returned true, and got 17 for getOwnershipVal(“O1”).

1. **function addOwner(bytes32 addr, uint val) returns (bool txnPassed)**

**function removeOwner(bytes32 addr) returns (bool txnPassed)  
function isOwner(bytes32 addr) returns (bool isOwner)**

These functions were verified by first calling addOwner(“O5”,3), looking at the arrays ownershipVals and ownershipHashes (made public for testing),calling isOwner(“O5”), then calling removeOwner(“O5”, and again looking at ownershipVals and ownershipHashes, then checking isOwner(“O5”). Additionally, both functions addOwner and removeOwner returned true.

1. **function getOwnersList() returns (bytes32[100] ownersList)** was validated in **(2)**.