CertiK Verification Report for ContentOS

Verification Request date: 2018-XX-XX
Company Name: https://www.contentos.io

Summary

This is the report for smart contract verification service on contentos.sol from ContentOS. The goal of the audition is to guarantee that verified smart contracts are robstic enough to avoid potentially unexpected loopholes.
You can find the source code at
https://etherscan.io/address/0x589891a198195061cb8ad1a75357a3b7dbadd7bc#code

Conclusion: PASS

Given the fact that all property checkings and partially function correctness were applied to the source code, our formal verification engine concludes that the ContentOS Smart Contract implementation meets 100% of the specification, based on the 100% code coverage. CertiK believes the contracts are mostly trustworthy and hack-resistant.

Details

1. Vulnerability

CertiK will apply 100% coveraged smart labels on the source code to detect 4 types of errors: Function Correctness, Integer Overflow, Assertion Failure, Array Index out of Bound. For each failed verification request, CertiK will base on its severity to push to 3 buckets:

Critical, Medium and Low. Other than issues falling into Low level, CertiK will push back and require the client to update the source code to meet criteria.

Critical:

NOT FOUND

Major (Medium):

NOT FOUND

Minor(Low):

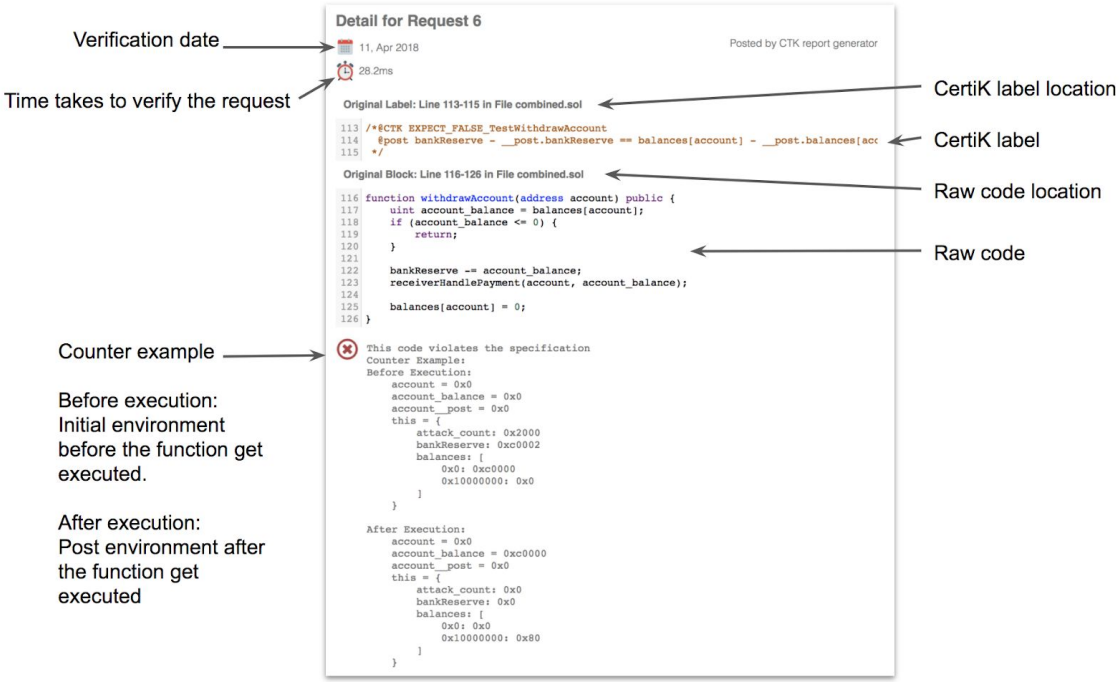
NOT FOUND

2. Other suggestions

CertiK will also provide suggestions based on the consideration of System Design, Algorithm Design, Gas Optimization and Library Using.

a. In **COSTokenBase**, the **constructor** may have **Integer Overflow** in **totalSupply = _initialSupply * 10 ** uint256(decimals)**, if other child is using this. It will be better to add **require** to bound "*_initialSupply*" and "*_decimals*"
b. For all the transfer functions, it is better to add **require(_from != to)** or **require(msg.sender != to)**, it can prevent the gas paid by transfering to the user himself.
c. **require(msg.sender == owner)** inside **transferOfPower** is a redundant precondition with modifier **auth.**
d. Could create a SafeMath library with add and sub function, to replace the overflow check in those function with "+" and "-".

3. How to Read



4. Disclaimer

This Report is subject to the terms and conditions (including without limitation, description of the services, confidentiality, disclaimer and limitation of liability) set forth in the Verification Services Agreement between CertiK and the Client, or the scope of verification, and terms and conditions provided to the Client in connection with this verification. No third party shall be entitled to rely on this Report or have any legal or equitable right, benefit or remedy of any nature whatsoever, under or by reason of this Report. CertiK assumes no liability to any third party because of reliance on this Report.

CertiK

Certi Request Report

98 out of 98 specs are satisfied.

Detail for Request 0: If method completes, integer overflow would not happen.

15, Jul 2018

Posted by CTK report generatc

1.5ms

Line 77 in File contentos.sol

```
77 //@CTK NO_OVERFLOW
```

Line 85-87 in File contentos.sol

```
85 function stop() auth internal {
86     stopped = true;
87 }
```

The code meets the specification

Detail for Request 1: Buffer overflow / array index out of bound would never happen.

15, Jul 2018

Posted by CTK report generatc

1ms

Line 78 in File contentos.sol

```
78 //@CTK NO_BUF_OVERFLOW
```

Line 85-87 in File contentos.sol

```
85 function stop() auth internal {
86     stopped = true;
87 }
```

The code meets the specification

Detail for Request 2: Method will not encounter an assertion failure.

15, Jul 2018

Posted by CTK report generatc

1.3ms

Line 79 in File contentos.sol

```
79 //@CTK NO_ASF
```

Line 85-87 in File contentos.sol

```
85 function stop() auth internal {
86     stopped = true;
87 }
```

The code meets the specification

Detail for Request 3: stop

15, Jul 2018

Posted by CTK report generatc

6.6ms

Line 80-84 in File contentos.sol

```
80 /*@CTK "stop"
81   @tag assume_completion
82   @post owner == msg.sender
83   @post __post.stopped == true
84 */
```

Line 85-87 in File contentos.sol

```
85 function stop() auth internal {
86     stopped = true;
87 }
```

The code meets the specification

Detail for Request 4: If method completes, integer overflow would not happen.

15, Jul 2018

Posted by CTK report generatc

1.3ms

Line 88 in File contentos.sol

```
88 //@CTK NO_OVERFLOW
```

Line 96-98 in File contentos.sol

```
96 function start() auth internal {
97     stopped = false;
98 }
```

The code meets the specification

Detail for Request 5: Buffer overflow / array index out of bound would never happen.

15, Jul 2018

Posted by CTK report generatc

1ms

Line 89 in File contentos.sol

```
89 //@CTK NO_BUF_OVERFLOW
```

Line 96-98 in File contentos.sol

```
96 function start() auth internal {
97     stopped = false;
98 }
```

The code meets the specification

Detail for Request 6: Method will not encounter an assertion failure.

15, Jul 2018

Posted by CTK report generatc

0.9ms

Line 90 in File contentos.sol

```
90 //@CTK NO_ASF
```

Line 96-98 in File contentos.sol

```
96 function start() auth internal {
97     stopped = false;
98 }
```

The code meets the specification

Detail for Request 7: start

15, Jul 2018

Posted by CTK report generatc

4.4ms

Line 91-95 in File contentos.sol

```
91 /*@CTK "start"
92   @tag assume_completion
93   @post owner == msg.sender
94   @post __post.stopped == false
95 */
```

Line 96-98 in File contentos.sol

```
96 function start() auth internal {
97     stopped = false;
98 }
```

The code meets the specification

Detail for Request 8: If method completes, integer overflow would not happen.

15, Jul 2018

Posted by CTK report generatc

0.7ms

Line 591 in File contentos.sol

```
591 //@CTK NO_OVERFLOW
```

Line 602-607 in File contentos.sol

```
602     function register(string key) public {
603 //        require(bytes(key).length <= 64);
604         require(balances[msg.sender] > 0);
605         register_map[msg.sender] = key;
606         emit LogRegister(msg.sender, key);
607     }
```

The code meets the specification

Detail for Request 9: Buffer overflow / array index out of bound would never happen.

15, Jul 2018

Posted by CTK report generatc

0.7ms

Line 592 in File contentos.sol

```
592 //@CTK NO_BUF_OVERFLOW
```

Line 602-607 in File contentos.sol

```
602     function register(string key) public {
603 //        require(bytes(key).length <= 64);
604         require(balances[msg.sender] > 0);
605         register_map[msg.sender] = key;
606         emit LogRegister(msg.sender, key);
607     }
```

The code meets the specification

Detail for Request 10: Method will not encounter an assertion failure.

15, Jul 2018

Posted by CTK report generatc

0.7ms

Line 593 in File contentos.sol

```
593 //@CTK NO_ASF
```

Line 602-607 in File contentos.sol

```
602     function register(string key) public {
603 //        require(bytes(key).length <= 64);
604         require(balances[msg.sender] > 0);
605         register_map[msg.sender] = key;
606         emit LogRegister(msg.sender, key);
607     }
```

The code meets the specification

Detail for Request 11: decreaseApproval prerequisite fail

15, Jul 2018

Posted by CTK report generatc

2.6ms

Line 594-596 in File contentos.sol

```
594 /*@CTK "decreaseApproval prerequisite fail"
595   @post (balances[msg.sender] <= 0) -> __reverted == true
596 */
```

Line 602-607 in File contentos.sol

```
602     function register(string key) public {
603 //        require(bytes(key).length <= 64);
604         require(balances[msg.sender] > 0);
605         register_map[msg.sender] = key;
606         emit LogRegister(msg.sender, key);
607     }
```

The code meets the specification

Detail for Request 12: register

15, Jul 2018

Posted by CTK report generatc

5.4ms

Line 597-601 in File contentos.sol

```
597 /*@CTK "register"
598   @tag assume_completion
599   @pre balances[msg.sender] > 0
600   @post __post.register_map[msg.sender] == key
601 */
```

Line 602-607 in File contentos.sol

```
602     function register(string key) public {
603 //        require(bytes(key).length <= 64);
604         require(balances[msg.sender] > 0);
605         register_map[msg.sender] = key;
606         emit LogRegister(msg.sender, key);
607     }
```

The code meets the specification

Detail for Request 13: Buffer overflow / array index out of bound would never happen.

15, Jul 2018

Posted by CTK report generatc

142.3ms

Line 274 in File contentos.sol

```
274 //@CTK NO_BUF_OVERFLOW
```

Line 291-293 in File contentos.sol

```
291 function transfer(address _to, uint256 _value) stoppable public returns(bool) {
292     return _transfer(msg.sender, _to, _value);
293 }
```

The code meets the specification

Detail for Request 14: Method will not encounter an assertion failure.

15, Jul 2018

Posted by CTK report generatc

714ms

Line 275 in File contentos.sol

```
275 //@CTK NO_ASF
```

Line 291-293 in File contentos.sol

```
291 function transfer(address _to, uint256 _value) stoppable public returns(bool) {
292     return _transfer(msg.sender, _to, _value);
293 }
```

The code meets the specification

Detail for Request 15: transfer2

15, Jul 2018

Posted by CTK report generatc

861.9ms

Line 276-290 in File contentos.sol

```
276 /*@CTK "transfer2"
277   @tag assume_completion
278   @pre _to != 0x0
279   @pre balances[msg.sender] >= _value
280   @pre balances[_to] + _value > balances[_to]
281   @pre balances[_to] + balances[msg.sender] >= balances[_to]
282   @pre _freezeList[msg.sender] == false
283   @pre stopped == false
284   @post !__has_overflow
285   @post msg.sender != _to -> __post.balances[_to] == balances[_to] + _value
286   @post msg.sender != _to -> __post.balances[msg.sender] == balances[msg.sender] -
287   @post msg.sender == _to -> __post.balances[_to] == balances[_to]
288   @post msg.sender == _to -> __post.balances[msg.sender] == balances[msg.sender]
289   @post __return == true
290 */
```

Line 291-293 in File contentos.sol

```
291 function transfer(address _to, uint256 _value) stoppable public returns(bool) {
292     return _transfer(msg.sender, _to, _value);
293 }
```

The code meets the specification

Detail for Request 16: If method completes, integer overflow would not happen.

15, Jul 2018

Posted by CTK report generatc

124.4ms

Line 417 in File contentos.sol

```
417 //@CTK NO_OVERFLOW
```

Line 431-437 in File contentos.sol

```
431 function mint(uint256 _value) auth stoppable public returns(bool){
432     require(balances[msg.sender] + _value > balances[msg.sender]);
433     require(totalSupply + _value > totalSupply);
434     balances[msg.sender] += _value;
435     totalSupply += _value;
436     return true;
437 }
```

The code meets the specification

Detail for Request 17: Buffer overflow / array index out of bound would never happen.

15, Jul 2018

Posted by CTK report generatc

15.3ms

Line 418 in File contentos.sol

```
418 //@CTK NO_BUF_OVERFLOW
```

Line 431-437 in File contentos.sol

```
431 function mint(uint256 _value) auth stoppable public returns(bool){
```

```
432     require(balances[msg.sender] + _value > balances[msg.sender]);
433     require(totalSupply + _value > totalSupply);
434     balances[msg.sender] += _value;
435     totalSupply += _value;
436     return true;
437 }
```

The code meets the specification

Detail for Request 18: Method will not encounter an assertion failure.

15, Jul 2018

Posted by CTK report generatc

28.2ms

Line 419 in File contentos.sol

```
419 //@CTK NO_ASF
```

Line 431-437 in File contentos.sol

```
431 function mint(uint256 _value) auth stoppable public returns(bool){
432     require(balances[msg.sender] + _value > balances[msg.sender]);
433     require(totalSupply + _value > totalSupply);
434     balances[msg.sender] += _value;
435     totalSupply += _value;
436     return true;
437 }
```

The code meets the specification

Detail for Request 19: mint prerequisite fail

15, Jul 2018

Posted by CTK report generatc

17.9ms

Line 420-422 in File contentos.sol

```
420 /*@CTK "mint prerequisite fail"
```

```
421   @post (stopped == true) \/ (msg.sender != owner) -> __reverted == true
422 */
```

Line 431-437 in File contentos.sol

```
431 function mint(uint256 _value) auth stoppable public returns(bool){
432     require(balances[msg.sender] + _value > balances[msg.sender]);
433     require(totalSupply + _value > totalSupply);
434     balances[msg.sender] += _value;
435     totalSupply += _value;
436     return true;
437 }
```

The code meets the specification

Detail for Request 20: mint

15, Jul 2018

Posted by CTK report generatc

352.2ms

Line 423-430 in File contentos.sol

```
423 /*@CTK "mint"
424   @tag assume_completion
425   @pre stopped == false
426   @pre owner == msg.sender
427   @post __post.balances[msg.sender] == balances[msg.sender] + _value
428   @post __post.totalSupply == totalSupply + _value
429   @post __return == true
430 */
```

Line 431-437 in File contentos.sol

```
431 function mint(uint256 _value) auth stoppable public returns(bool){
432     require(balances[msg.sender] + _value > balances[msg.sender]);
433     require(totalSupply + _value > totalSupply);
434     balances[msg.sender] += _value;
435     totalSupply += _value;
436     return true;
437 }
```

The code meets the specification

Detail for Request 21: If method completes, integer overflow would not happen.

15, Jul 2018

Posted by CTK report generatc

52.5ms

Line 513 in File contentos.sol

```
513 //@CTK NO_OVERFLOW
```

Line 525-531 in File contentos.sol

```
525 function increaseApproval(address _spender, uint _addedValue) stoppable public ret
526     // Check for overflows
527     require(allowances[msg.sender][_spender] + _addedValue > allowances[msg.sender
528     allowances[msg.sender][_spender] += _addedValue;
529     emit Approval(msg.sender, _spender, allowances[msg.sender][_spender]);
530     return true;
531 }
```

The code meets the specification

Detail for Request 22: Buffer overflow / array index out of bound would never happen.

15, Jul 2018

Posted by CTK report generatc

1.8ms

Line 514 in File contentos.sol

```
514 //@CTK NO_BUF_OVERFLOW
```

Line 525-531 in File contentos.sol

```
525 function increaseApproval(address _spender, uint _addedValue) stoppable public ret
526     // Check for overflows
527     require(allowances[msg.sender][_spender] + _addedValue > allowances[msg.sender
528     allowances[msg.sender][_spender] += _addedValue;
529     emit Approval(msg.sender, _spender, allowances[msg.sender][_spender]);
530     return true;
531 }
```

The code meets the specification

Detail for Request 23: Method will not encounter an assertion failure.

15, Jul 2018

Posted by CTK report generatc

1.6ms

Line 515 in File contentos.sol

```
515 //@CTK NO_ASF
```

Line 525-531 in File contentos.sol

```
525 function increaseApproval(address _spender, uint _addedValue) stoppable public ret
526     // Check for overflows
527     require(allowances[msg.sender][_spender] + _addedValue > allowances[msg.sender
528     allowances[msg.sender][_spender] += _addedValue;
529     emit Approval(msg.sender, _spender, allowances[msg.sender][_spender]);
530     return true;
531 }
```

The code meets the specification

Detail for Request 24: increaseApproval prerequisite fail

15, Jul 2018

Posted by CTK report generatc

9ms

Line 516-518 in File contentos.sol

```
516 /*@CTK "increaseApproval prerequisite fail"
517   @post (stopped == true) \/ (allowances[msg.sender][_spender] + _addedValue <= al
518 */
```

Line 525-531 in File contentos.sol

```
525 function increaseApproval(address _spender, uint _addedValue) stoppable public ret
526     // Check for overflows
527     require(allowances[msg.sender][_spender] + _addedValue > allowances[msg.sender
528     allowances[msg.sender][_spender] += _addedValue;
529     emit Approval(msg.sender, _spender, allowances[msg.sender][_spender]);
530     return true;
531 }
```

The code meets the specification

Detail for Request 25: increaseApproval

15, Jul 2018

Posted by CTK report generatc

76.7ms

Line 519-524 in File contentos.sol

```
519 /*@CTK "increaseApproval"
520   @pre stopped == false
521   @pre allowances[msg.sender][_spender] + _addedValue > allowances[msg.sender][_sp
522   @post __post.allowances[msg.sender][_spender] == allowances[msg.sender][_spender
523   @post __return == true
524 */
```

Line 525-531 in File contentos.sol

```
525 function increaseApproval(address _spender, uint _addedValue) stoppable public ret
526     // Check for overflows
527     require(allowances[msg.sender][_spender] + _addedValue > allowances[msg.sender
528     allowances[msg.sender][_spender] += _addedValue;
529     emit Approval(msg.sender, _spender, allowances[msg.sender][_spender]);
530     return true;
531 }
```

The code meets the specification

Detail for Request 26: If method completes, integer overflow would not happen.

15, Jul 2018

Posted by CTK report generatc

0.8ms

Line 54 in File contentos.sol

```
54 //@CTK NO_OVERFLOW
```

Line 60-66 in File contentos.sol

```
60 function isAuthorized(address src) internal view returns (bool) {
61     if(src == owner){
62         return true;
63     } else {
64         return false;
65     }
66 }
```

The code meets the specification

Detail for Request 27: Buffer overflow / array index out of bound would never happen.

15, Jul 2018

Posted by CTK report generatc

0.7ms

Line 55 in File contentos.sol

```
55 //@CTK NO_BUF_OVERFLOW
```

Line 60-66 in File contentos.sol

```
60 function isAuthorized(address src) internal view returns (bool) {
61     if(src == owner){
62         return true;
63     } else {
64         return false;
65     }
66 }
```

The code meets the specification

Detail for Request 28: Method will not encounter an assertion failure.

15, Jul 2018

Posted by CTK report generatc

0.6ms

Line 56 in File contentos.sol

```
56 //@CTK NO_ASF
```

Line 60-66 in File contentos.sol

```
60 function isAuthorized(address src) internal view returns (bool) {
61     if(src == owner){
62         return true;
63     } else {
64         return false;
65     }
66 }
```

The code meets the specification

Detail for Request 29: isAuthorized

15, Jul 2018

Posted by CTK report generatc

5ms

Line 57-59 in File contentos.sol

```
57 /*@CTK isAuthorized
58   @post __return == (src == owner)
59 */
```

Line 60-66 in File contentos.sol

```
60 function isAuthorized(address src) internal view returns (bool) {
61     if(src == owner){
62         return true;
63     } else {
64         return false;
65     }
66 }
```

The code meets the specification

Detail for Request 30: If method completes, integer overflow would not happen.

15, Jul 2018

Posted by CTK report generatc

2.1ms

Line 129 in File contentos.sol

```
129 //@CTK NO_OVERFLOW
```

Line 138-145 in File contentos.sol

```
138 function unfreeze(address addr) auth public returns (bool) {
139   require(true == _freezeList[addr]);
140 
141   _freezeList[addr] = false;
142 
143   emit UnFreezed(addr);
144   return true;
145 }
```

The code meets the specification

Detail for Request 31: Buffer overflow / array index out of bound would never happen.

15, Jul 2018

Posted by CTK report generatc

1.4ms

Line 130 in File contentos.sol

```
130 //@CTK NO_BUF_OVERFLOW
```

Line 138-145 in File contentos.sol

```
138 function unfreeze(address addr) auth public returns (bool) {
139   require(true == _freezeList[addr]);
140 
141   _freezeList[addr] = false;
142 
143   emit UnFreezed(addr);
144   return true;
145 }
```

The code meets the specification

Detail for Request 32: Method will not encounter an assertion failure.

15, Jul 2018

Posted by CTK report generatc

2ms

Line 131 in File contentos.sol

```
131 //@CTK NO_ASF
```

Line 138-145 in File contentos.sol

```
138 function unfreeze(address addr) auth public returns (bool) {
139   require(true == _freezeList[addr]);
140
141   _freezeList[addr] = false;
142
143   emit UnFreezed(addr);
144   return true;
145 }
```

The code meets the specification

Detail for Request 33: unfreeze

15, Jul 2018

Posted by CTK report generatc

10ms

Line 132-137 in File contentos.sol

```
132 /*@CTK "unfreeze"
133   @tag assume_completion
134   @pre _freezeList[addr] == true
135   @post __post._freezeList[addr] == false
136   @post __return == true
137 */
```

Line 138-145 in File contentos.sol

```
138 function unfreeze(address addr) auth public returns (bool) {
139   require(true == _freezeList[addr]);
140
141   _freezeList[addr] = false;
142
143   emit UnFreezed(addr);
144   return true;
145 }
```

The code meets the specification

Detail for Request 34: If method completes, integer overflow would not happen.

15, Jul 2018

Posted by CTK report generatc

1.5ms

Line 578 in File contentos.sol

```
578 //@CTK NO_OVERFLOW
```

Line 586-589 in File contentos.sol

```
586 function finish() public{
587     stop();
588     emit LogStop();
589 }
```

The code meets the specification

Detail for Request 35: Buffer overflow / array index out of bound would never happen.

15, Jul 2018

Posted by CTK report generatc

1.6ms

Line 579 in File contentos.sol

```
579 //@CTK NO_BUF_OVERFLOW
```

Line 586-589 in File contentos.sol

```
586 function finish() public{
587     stop();
588     emit LogStop();
589 }
```

The code meets the specification

Detail for Request 36: Method will not encounter an assertion failure.

15, Jul 2018

Posted by CTK report generatc

1.3ms

Line 580 in File contentos.sol

```
580 //@CTK NO_ASF
```

Line 586-589 in File contentos.sol

```
586 function finish() public{
587     stop();
588     emit LogStop();
589 }
```

The code meets the specification

Detail for Request 37: finish

15, Jul 2018

Posted by CTK report generatc

7.1ms

Line 581-585 in File contentos.sol

```
581 /*@CTK "finish"
582   @tag assume_completion
583   @post owner == msg.sender
584   @post __post.stopped == true
585 */
```

Line 586-589 in File contentos.sol

```
586 function finish() public{
587     stop();
588     emit LogStop();
589 }
```

The code meets the specification

Detail for Request 38: If method completes, integer overflow would not happen.

15, Jul 2018

Posted by CTK report generatc

0.7ms

Line 152 in File contentos.sol

```
152 //@CTK NO_OVERFLOW
```

Line 158-164 in File contentos.sol

```
158 function isFreezing(address addr) public view returns (bool) {
159     if (true == _freezeList[addr]) {
160         return true;
161     } else {
162         return false;
163     }
164 }
```

The code meets the specification

Detail for Request 39: Buffer overflow / array index out of bound would never happen.

15, Jul 2018

Posted by CTK report generatc

0.6ms

Line 153 in File contentos.sol

```
153 //@CTK NO_BUF_OVERFLOW
```

Line 158-164 in File contentos.sol

```
158 function isFreezing(address addr) public view returns (bool) {
159     if (true == _freezeList[addr]) {
160         return true;
161     } else {
162         return false;
163     }
164 }
```

The code meets the specification

Detail for Request 40: Method will not encounter an assertion failure.

15, Jul 2018

Posted by CTK report generatc

0.4ms

Line 154 in File contentos.sol

```
154 //@CTK NO_ASF
```

Line 158-164 in File contentos.sol

```
158 function isFreezing(address addr) public view returns (bool) {
159     if (true == _freezeList[addr]) {
160         return true;
161     } else {
162         return false;
163     }
164 }
```

The code meets the specification

Detail for Request 41: isFreezing

15, Jul 2018

Posted by CTK report generatc

5.8ms

Line 155-157 in File contentos.sol

```
155 /*@CTK "isFreezing"
156   @post __return == _freezeList[addr]
157 */
```

Line 158-164 in File contentos.sol

```
158 function isFreezing(address addr) public view returns (bool) {
159     if (true == _freezeList[addr]) {
160         return true;
161     } else {
162         return false;
163     }
164 }
```

The code meets the specification

Detail for Request 42: Buffer overflow / array index out of bound would never happen.

15, Jul 2018

Posted by CTK report generatc

1.4ms

Line 387 in File contentos.sol

```
387 //@CTK NO_BUF_OVERFLOW
```

Line 402-408 in File contentos.sol

```
402 function burn(uint256 _value) stoppable public returns(bool)  {
403     require(balances[msg.sender] >= _value);   // Check if the sender has enough
404     balances[msg.sender] -= _value;            // Subtract from the sender
405     totalSupply -= _value;                      // Updates totalSupply
406     emit Burn(msg.sender, _value);
407     return true;
408 }
```

The code meets the specification

Detail for Request 43: Method will not encounter an assertion failure.

15, Jul 2018

Posted by CTK report generatc

1.5ms

Line 388 in File contentos.sol

```
388 //@CTK NO_ASF
```

Line 402-408 in File contentos.sol

```
402 function burn(uint256 _value) stoppable public returns(bool)  {
403     require(balances[msg.sender] >= _value);   // Check if the sender has enough
404     balances[msg.sender] -= _value;            // Subtract from the sender
405     totalSupply -= _value;                      // Updates totalSupply
406     emit Burn(msg.sender, _value);
407     return true;
408 }
```

The code meets the specification

Detail for Request 44: burn prerequisite fail

15, Jul 2018

Posted by CTK report generatc

9.9ms

Line 389-391 in File contentos.sol

```
389 /*@CTK "burn prerequisite fail"
390   @post (stopped == true) \/ (balances[msg.sender] < _value) -> __reverted == true
391 */
```

Line 402-408 in File contentos.sol

```
402 function burn(uint256 _value) stoppable public returns(bool)  {
403     require(balances[msg.sender] >= _value);   // Check if the sender has enough
404     balances[msg.sender] -= _value;            // Subtract from the sender
405     totalSupply -= _value;                      // Updates totalSupply
406     emit Burn(msg.sender, _value);
407     return true;
408 }
```

The code meets the specification

Detail for Request 45: burn

15, Jul 2018

Posted by CTK report generatc

395.7ms

Line 392-401 in File contentos.sol

```
392 /*@CTK "burn"
393   @tag assume_completion
394   @pre balances[msg.sender] >= _value
395   @pre balances[msg.sender] <= totalSupply
396   @pre stopped == false
397   @post !__has_overflow
398   @post __post.balances[msg.sender] == balances[msg.sender] - _value
399   @post __post.totalSupply == totalSupply - _value
400   @post __return == true
401 */
```

Line 402-408 in File contentos.sol

```
402 function burn(uint256 _value) stoppable public returns(bool)  {
403     require(balances[msg.sender] >= _value);   // Check if the sender has enough
404     balances[msg.sender] -= _value;            // Subtract from the sender
405     totalSupply -= _value;                      // Updates totalSupply
406     emit Burn(msg.sender, _value);
407     return true;
408 }
```

The code meets the specification

Detail for Request 46: Buffer overflow / array index out of bound would never happen.

15, Jul 2018

Posted by CTK report generatc

47.9ms

Line 304 in File contentos.sol

```
304 //@CTK NO_BUF_OVERFLOW
```

Line 323-327 in File contentos.sol

```
323 function transferFrom(address _from, address _to, uint256 _value) stoppable public
324     require(_value <= allowances[_from][msg.sender]);     // Check allowance
325     allowances[_from][msg.sender] -= _value;
326     return _transfer(_from, _to, _value);
327 }
```

The code meets the specification

Detail for Request 47: Method will not encounter an assertion failure.

15, Jul 2018

Posted by CTK report generatc

665.1ms

Line 305 in File contentos.sol

```
305 //@CTK NO_ASF
```

Line 323-327 in File contentos.sol

```
323 function transferFrom(address _from, address _to, uint256 _value) stoppable public
324     require(_value <= allowances[_from][msg.sender]);     // Check allowance
325     allowances[_from][msg.sender] -= _value;
326     return _transfer(_from, _to, _value);
327 }
```

The code meets the specification

Detail for Request 48: transferFrom

15, Jul 2018

Posted by CTK report generatc

1395.3ms

Line 306-322 in File contentos.sol

```
306 /*@CTK "transferFrom"
307   @tag assume_completion
308   @pre _to != 0x0
309   @pre allowances[_from][msg.sender] >= _value
310   @pre balances[msg.sender] >= _value
311   @pre balances[_to] + _value > balances[_to]
312   @pre balances[_to] + balances[_from] >= balances[_to]
313   @pre _freezeList[msg.sender] == false
314   @pre stopped == false
315   @post !__has_overflow
316   @post _from != _to -> __post.balances[_to] == balances[_to] + _value
317   @post _from != _to -> __post.balances[_from] == balances[_from] - _value
318   @post _from == _to -> __post.balances[_to] == balances[_to]
319   @post _from == _to -> __post.balances[_from] == balances[_from]
320   @post __post.allowances[_from][msg.sender] == allowances[_from][msg.sender] - _v
321   @post __return == true
322 */
```

Line 323-327 in File contentos.sol

```
323 function transferFrom(address _from, address _to, uint256 _value) stoppable public
324     require(_value <= allowances[_from][msg.sender]);     // Check allowance
325     allowances[_from][msg.sender] -= _value;
326     return _transfer(_from, _to, _value);
327 }
```

The code meets the specification

Detail for Request 49: If method completes, integer overflow would not happen.

15, Jul 2018

Posted by CTK report generatc

1.1ms

Line 337 in File contentos.sol

```
337 //@CTK NO_OVERFLOW
```

Line 348-352 in File contentos.sol

```
348 function approve(address _spender, uint256 _value) stoppable public returns(bool)
349     allowances[msg.sender][_spender] = _value;
350     emit Approval(msg.sender, _spender, _value);
351     return true;
352 }
```

The code meets the specification

Detail for Request 50: Buffer overflow / array index out of bound would never happen.

15, Jul 2018

Posted by CTK report generatc

1ms

Line 338 in File contentos.sol

```
338 //@CTK NO_BUF_OVERFLOW
```

Line 348-352 in File contentos.sol

```
348 function approve(address _spender, uint256 _value) stoppable public returns(bool)
349     allowances[msg.sender][_spender] = _value;
350     emit Approval(msg.sender, _spender, _value);
351     return true;
352 }
```

The code meets the specification

Detail for Request 51: Method will not encounter an assertion failure.

15, Jul 2018

Posted by CTK report generatc

1ms

Line 339 in File contentos.sol

```
339 //@CTK NO_ASF
```

Line 348-352 in File contentos.sol

```
348 function approve(address _spender, uint256 _value) stoppable public returns(bool)
349     allowances[msg.sender][_spender] = _value;
350     emit Approval(msg.sender, _spender, _value);
351     return true;
352 }
```

The code meets the specification

Detail for Request 52: approve prerequisite fail

15, Jul 2018

Posted by CTK report generatc

2.5ms

Line 340-342 in File contentos.sol

```
340 /*@CTK "approve prerequisite fail"
341   @post (stopped == true) -> __reverted == true
342 */
```

Line 348-352 in File contentos.sol

```
348 function approve(address _spender, uint256 _value) stoppable public returns(bool)
349     allowances[msg.sender][_spender] = _value;
350     emit Approval(msg.sender, _spender, _value);
351     return true;
352 }
```

The code meets the specification

Detail for Request 53: approve

15, Jul 2018

Posted by CTK report generatc

8.2ms

Line 343-347 in File contentos.sol

```
343 /*@CTK "approve"
344   @pre stopped == false
345   @post __post.allowances[msg.sender][_spender] == _value
346   @post __return == true
347 */
```

Line 348-352 in File contentos.sol

```
348 function approve(address _spender, uint256 _value) stoppable public returns(bool)
349     allowances[msg.sender][_spender] = _value;
```

```
350     emit Approval(msg.sender, _spender, _value);
351     return true;
352 }
```

The code meets the specification

Detail for Request 54: If method completes, integer overflow would not happen.

15, Jul 2018

Posted by CTK report generatc

2.9ms

Line 363 in File contentos.sol

```
363 //@CTK NO_OVERFLOW
```

Line 371-378 in File contentos.sol

```
371     function approveAndCall(address _spender, uint256 _value, bytes _extraData) st
372         if (approve(_spender, _value)) {
373             TokenRecipient spender = TokenRecipient(_spender);
374 //            spender.receiveApproval(msg.sender, _value, this, _extraData);
375             return true;
376         }
377         return false;
378     }
```

The code meets the specification

Detail for Request 55: Buffer overflow / array index out of bound would never happen.

15, Jul 2018

Posted by CTK report generatc

6.3ms

Line 364 in File contentos.sol

```
364 //@CTK NO_BUF_OVERFLOW
```

Line 371-378 in File contentos.sol

```
371     function approveAndCall(address _spender, uint256 _value, bytes _extraData) st
372         if (approve(_spender, _value)) {
373             TokenRecipient spender = TokenRecipient(_spender);
374 //            spender.receiveApproval(msg.sender, _value, this, _extraData);
375             return true;
376         }
377         return false;
378     }
```

The code meets the specification

Detail for Request 56: Method will not encounter an assertion failure.

15, Jul 2018

Posted by CTK report generatc

2.7ms

Line 365 in File contentos.sol

```
365 //@CTK NO_ASF
```

Line 371-378 in File contentos.sol

```
371     function approveAndCall(address _spender, uint256 _value, bytes _extraData) st
372         if (approve(_spender, _value)) {
373             TokenRecipient spender = TokenRecipient(_spender);
374 //            spender.receiveApproval(msg.sender, _value, this, _extraData);
375             return true;
376         }
377         return false;
378     }
```

The code meets the specification

Detail for Request 57: approveAndCall

Posted by CTK report generatc

15, Jul 2018
17.5ms

Line 366-370 in File contentos.sol

```
366 /*@CTK "approveAndCall"
367   @pre stopped == false
368   @post __post.allowances[msg.sender][_spender] == _value
369   @post __return == true
370 */
```

Line 371-378 in File contentos.sol

```
371     function approveAndCall(address _spender, uint256 _value, bytes _extraData) st
372         if (approve(_spender, _value)) {
373             TokenRecipient spender = TokenRecipient(_spender);
374 //            spender.receiveApproval(msg.sender, _value, this, _extraData);
375             return true;
376         }
377         return false;
378     }
```

The code meets the specification

Detail for Request 58: Buffer overflow / array index out of bound would never happen.

15, Jul 2018

Posted by CTK report generatc

10.3ms

Line 447 in File contentos.sol

```
447 //@CTK NO_BUF_OVERFLOW
```

Line 464-472 in File contentos.sol

```
464 function burnFrom(address _from, uint256 _value) stoppable public returns(bool) {
465     require(balances[_from] >= _value);                // Check if the targeted ba
466     require(_value <= allowances[_from][msg.sender]);    // Check allowance
467     balances[_from] -= _value;                         // Subtract from the target
468     allowances[_from][msg.sender] -= _value;             // Subtract from the send
469     totalSupply -= _value;                              // Update totalSupply
470     emit Burn(_from, _value);
471     return true;
472 }
```

The code meets the specification

Detail for Request 59: Method will not encounter an assertion failure.

15, Jul 2018

Posted by CTK report generatc

14.1ms

Line 448 in File contentos.sol

```
448 //@CTK NO_ASF
```

Line 464-472 in File contentos.sol

```
464 function burnFrom(address _from, uint256 _value) stoppable public returns(bool) {
465     require(balances[_from] >= _value);                 // Check if the targeted ba
466     require(_value <= allowances[_from][msg.sender]);    // Check allowance
467     balances[_from] -= _value;                          // Subtract from the target
468     allowances[_from][msg.sender] -= _value;             // Subtract from the senc
469     totalSupply -= _value;                               // Update totalSupply
470     emit Burn(_from, _value);
471     return true;
472 }
```

The code meets the specification

Detail for Request 60: burnFrom prerequisite fail

15, Jul 2018

Posted by CTK report generatc

101.6ms

Line 449-451 in File contentos.sol

```
449 /*@CTK "burnFrom prerequisite fail"
450   @post (stopped == true) \/ (balances[_from] < _value) \/ (allowances[_from][msg.
451 */
```

Line 464-472 in File contentos.sol

```
464 function burnFrom(address _from, uint256 _value) stoppable public returns(bool) {
465     require(balances[_from] >= _value);                 // Check if the targeted ba
466     require(_value <= allowances[_from][msg.sender]);    // Check allowance
467     balances[_from] -= _value;                          // Subtract from the target
```

```
468     allowances[_from][msg.sender] -= _value;             // Subtract from the send
469     totalSupply -= _value;                               // Update totalSupply
470     emit Burn(_from, _value);
471     return true;
472 }
```

The code meets the specification

Detail for Request 61: burnFrom

15, Jul 2018

Posted by CTK report generatc

832.6ms

Line 452-463 in File contentos.sol

```
452 /*@CTK "burnFrom"
453   @tag assume_completion
454   @pre balances[_from] >= _value
455   @pre balances[_from] <= totalSupply
456   @pre allowances[_from][msg.sender] >= _value
457   @pre stopped == false
458   @post !__has_overflow
459   @post __post.balances[_from] == balances[_from] - _value
460   @post __post.allowances[_from][msg.sender] == allowances[_from][msg.sender] - _v
461   @post __post.totalSupply == totalSupply - _value
462   @post __return == true
463 */
```

Line 464-472 in File contentos.sol

```
464 function burnFrom(address _from, uint256 _value) stoppable public returns(bool) {
465     require(balances[_from] >= _value);                // Check if the targeted ba
466     require(_value <= allowances[_from][msg.sender]);    // Check allowance
467     balances[_from] -= _value;                         // Subtract from the target
468     allowances[_from][msg.sender] -= _value;             // Subtract from the send
469     totalSupply -= _value;                             // Update totalSupply
470     emit Burn(_from, _value);
471     return true;
472 }
```

The code meets the specification

Detail for Request 62: If method completes, integer overflow would not happen.

15, Jul 2018

Posted by CTK report generatc

0.7ms

Line 207 in File contentos.sol

```
207 //@CTK NO_OVERFLOW
```

Line 213-215 in File contentos.sol

```
213 function balanceOf(address _owner) public view returns(uint256) {
214     return balances[_owner];
215 }
```

The code meets the specification

Detail for Request 63: Buffer overflow / array index out of bound would never happen.

15, Jul 2018

Posted by CTK report generatc

0.6ms

Line 208 in File contentos.sol

```
208 //@CTK NO_BUF_OVERFLOW
```

Line 213-215 in File contentos.sol

```
213 function balanceOf(address _owner) public view returns(uint256) {
214     return balances[_owner];
215 }
```

The code meets the specification

Detail for Request 64: Method will not encounter an assertion failure.

15, Jul 2018

Posted by CTK report generatc

0.5ms

Line 209 in File contentos.sol

```
209 //@CTK NO_ASF
```

Line 213-215 in File contentos.sol

```
213 function balanceOf(address _owner) public view returns(uint256) {
214     return balances[_owner];
215 }
```

The code meets the specification

Detail for Request 65: balanceOf

15, Jul 2018

Posted by CTK report generatc

1.7ms

Line 210-212 in File contentos.sol

```
210 /*@CTK "balanceOf"
211   @post __return == balances[_owner]
212 */
```

Line 213-215 in File contentos.sol

```
213 function balanceOf(address _owner) public view returns(uint256) {
214     return balances[_owner];
215 }
```

The code meets the specification

Detail for Request 66: Buffer overflow / array index out of bound would never happen.

15, Jul 2018

Posted by CTK report generatc

38.7ms

Line 479 in File contentos.sol

```
479 //@CTK NO_BUF_OVERFLOW
```

Line 498-505 in File contentos.sol

```
498 function transferOfPower(address _to) auth stoppable public returns (bool) {
499     require(msg.sender == owner);
500     uint value = balances[msg.sender];
501     _transfer(msg.sender, _to, value);
502     owner = _to;
503     emit TransferOfPower(msg.sender, _to);
504     return true;
505 }
```

The code meets the specification

Detail for Request 67: Method will not encounter an assertion failure.

15, Jul 2018

Posted by CTK report generatc

478.6ms

Line 480 in File contentos.sol

```
480 //@CTK NO_ASF
```

Line 498-505 in File contentos.sol

```
498 function transferOfPower(address _to) auth stoppable public returns (bool) {
499     require(msg.sender == owner);
500     uint value = balances[msg.sender];
501     _transfer(msg.sender, _to, value);
502     owner = _to;
503     emit TransferOfPower(msg.sender, _to);
504     return true;
505 }
```

The code meets the specification

Detail for Request 68: transferOfPower prerequisite fail

15, Jul 2018

Posted by CTK report generatc

34.9ms

Line 481-483 in File contentos.sol

```
481 /*@CTK "transferOfPower prerequisite fail"
482   @post (stopped == true) \/ (msg.sender != owner) -> __reverted == true
483 */
```

Line 498-505 in File contentos.sol

```
498 function transferOfPower(address _to) auth stoppable public returns (bool) {
499     require(msg.sender == owner);
500     uint value = balances[msg.sender];
501     _transfer(msg.sender, _to, value);
502     owner = _to;
503     emit TransferOfPower(msg.sender, _to);
504     return true;
505 }
```

The code meets the specification

Detail for Request 69: transferOfPower

15, Jul 2018

Posted by CTK report generatc

449.6ms

Line 484-497 in File contentos.sol

```
484 /*@CTK "transferOfPower"
485   @tag assume_completion
486   @pre stopped == false
487   @pre owner == msg.sender
488   @pre _to != 0x0
489   @pre balances[_to] + balances[msg.sender] <= totalSupply
490   @pre balances[_to] + balances[msg.sender] >= balances[_to]
491   @pre _freezeList[msg.sender] == false
492   @post !__has_overflow
493   @post __post.owner == _to
494   @post msg.sender != _to -> __post.balances[_to] == balances[_to] + balances[msg.
```

```
  @post msg.sender == _to -> __post.balances[_to] == balances[_to]
  @post __return == true
*/
```

Line 498-505 in File contentos.sol

```
function transferOfPower(address _to) auth stoppable public returns (bool) {
    require(msg.sender == owner);
    uint value = balances[msg.sender];
    _transfer(msg.sender, _to, value);
    owner = _to;
    emit TransferOfPower(msg.sender, _to);
    return true;
}
```

The code meets the specification

Detail for Request 70: If method completes, integer overflow would not happen.

15, Jul 2018

Posted by CTK report generatc

2.2ms

Line 111 in File contentos.sol

```
//@CTK NO_OVERFLOW
```

Line 120-127 in File contentos.sol

```
function freeze(address addr) auth public returns (bool) {
  require(true != _freezeList[addr]);

  _freezeList[addr] = true;

  emit Freezed(addr);
  return true;
}
```

The code meets the specification

Detail for Request 71: Buffer overflow / array index out of bound would never happen.

15, Jul 2018

Posted by CTK report generatc

2.1ms

Line 112 in File contentos.sol

```
112 //@CTK NO_BUF_OVERFLOW
```

Line 120-127 in File contentos.sol

```
120 function freeze(address addr) auth public returns (bool) {
121   require(true != _freezeList[addr]);
122
123   _freezeList[addr] = true;
124
125   emit Freezed(addr);
126   return true;
127 }
```

The code meets the specification

Detail for Request 72: Method will not encounter an assertion failure.

15, Jul 2018

Posted by CTK report generatc

2.5ms

Line 113 in File contentos.sol

```
113 //@CTK NO_ASF
```

Line 120-127 in File contentos.sol

```
120 function freeze(address addr) auth public returns (bool) {
121   require(true != _freezeList[addr]);
122
123   _freezeList[addr] = true;
124
125   emit Freezed(addr);
126   return true;
127 }
```

The code meets the specification

Detail for Request 73: freeze

15, Jul 2018

Posted by CTK report generatc

11.2ms

Line 114-119 in File contentos.sol

```
114 /*@CTK "freeze"
115   @tag assume_completion
116   @pre _freezeList[addr] != true
117   @post __post._freezeList[addr] == true
118   @post __return == true
119 */
```

Line 120-127 in File contentos.sol

```
120 function freeze(address addr) auth public returns (bool) {
121   require(true != _freezeList[addr]);
122 
123   _freezeList[addr] = true;
124 
125   emit Freezed(addr);
126   return true;
127 }
```

The code meets the specification

Detail for Request 74: If method completes, integer overflow would not happen.

15, Jul 2018

Posted by CTK report generatc

0.5ms

Line 39 in File contentos.sol

```
39 //@CTK NO_OVERFLOW
```

Line 45-47 in File contentos.sol

```
45 constructor () public {
46      owner = msg.sender;
47 }
```

The code meets the specification

Detail for Request 75: Buffer overflow / array index out of bound would never happen.

15, Jul 2018

Posted by CTK report generatc

0.4ms

Line 40 in File contentos.sol

```
40 //@CTK NO_BUF_OVERFLOW
```

Line 45-47 in File contentos.sol

```
45 constructor () public {
46      owner = msg.sender;
47 }
```

The code meets the specification

Detail for Request 76: Method will not encounter an assertion failure.

15, Jul 2018

Posted by CTK report generatc

0.5ms

Line 41 in File contentos.sol

```
41 //@CTK NO_ASF
```

Line 45-47 in File contentos.sol

```
45 constructor () public {
46      owner = msg.sender;
47 }
```

The code meets the specification

Detail for Request 77: constructor

15, Jul 2018

Posted by CTK report generatc

3.9ms

Line 42-44 in File contentos.sol

```
42 /*@CTK "constructor"
43   @post __post.owner == msg.sender
44 */
```

Line 45-47 in File contentos.sol

```
45 constructor () public {
46      owner = msg.sender;
47 }
```

The code meets the specification

Detail for Request 78: If method completes, integer overflow would not happen.

15, Jul 2018

Posted by CTK report generatc

3.6ms

Line 533 in File contentos.sol

```
533 //@CTK NO_OVERFLOW
```

Line 551-560 in File contentos.sol

```
551 function decreaseApproval(address _spender, uint _subtractedValue) stoppable publi
552     uint oldValue = allowances[msg.sender][_spender];
553     if (_subtractedValue > oldValue) {
554         allowances[msg.sender][_spender] = 0;
```

```
555     } else {
556         allowances[msg.sender][_spender] = oldValue - _subtractedValue;
557     }
558     emit Approval(msg.sender, _spender, allowances[msg.sender][_spender]);
559     return true;
560 }
```

The code meets the specification

Detail for Request 79: Buffer overflow / array index out of bound would never happen.

15, Jul 2018

Posted by CTK report generatc

1.3ms

Line 534 in File contentos.sol

```
534 //@CTK NO_BUF_OVERFLOW
```

Line 551-560 in File contentos.sol

```
551 function decreaseApproval(address _spender, uint _subtractedValue) stoppable publi
552     uint oldValue = allowances[msg.sender][_spender];
553     if (_subtractedValue > oldValue) {
554         allowances[msg.sender][_spender] = 0;
555     } else {
556         allowances[msg.sender][_spender] = oldValue - _subtractedValue;
557     }
558     emit Approval(msg.sender, _spender, allowances[msg.sender][_spender]);
559     return true;
560 }
```

The code meets the specification

Detail for Request 80: Method will not encounter an assertion failure.

15, Jul 2018

Posted by CTK report generatc

1.2ms

Line 535 in File contentos.sol

```
535 //@CTK NO_ASF
```

Line 551-560 in File contentos.sol

```
551 function decreaseApproval(address _spender, uint _subtractedValue) stoppable publi
552     uint oldValue = allowances[msg.sender][_spender];
553     if (_subtractedValue > oldValue) {
554         allowances[msg.sender][_spender] = 0;
555     } else {
556         allowances[msg.sender][_spender] = oldValue - _subtractedValue;
557     }
558     emit Approval(msg.sender, _spender, allowances[msg.sender][_spender]);
559     return true;
560 }
```

The code meets the specification

Detail for Request 81: decreaseApproval prerequisite fail

15, Jul 2018

Posted by CTK report generatc

3.1ms

Line 536-538 in File contentos.sol

```
536 /*@CTK "decreaseApproval prerequisite fail"
537   @post (stopped == true) -> __reverted == true
538 */
```

Line 551-560 in File contentos.sol

```
551 function decreaseApproval(address _spender, uint _subtractedValue) stoppable publi
552     uint oldValue = allowances[msg.sender][_spender];
553     if (_subtractedValue > oldValue) {
554         allowances[msg.sender][_spender] = 0;
555     } else {
556         allowances[msg.sender][_spender] = oldValue - _subtractedValue;
557     }
558     emit Approval(msg.sender, _spender, allowances[msg.sender][_spender]);
559     return true;
560 }
```

The code meets the specification

Detail for Request 82: decreaseApproval case 1

15, Jul 2018

Posted by CTK report generatc

48.3ms

Line 539-544 in File contentos.sol

```
539 /*@CTK "decreaseApproval case 1"
540   @pre stopped == false
541   @pre allowances[msg.sender][_spender] < _subtractedValue
542   @post __post.allowances[msg.sender][_spender] == 0
543   @post __return == true
544 */
```

Line 551-560 in File contentos.sol

```
551 function decreaseApproval(address _spender, uint _subtractedValue) stoppable publi
552     uint oldValue = allowances[msg.sender][_spender];
553     if (_subtractedValue > oldValue) {
554         allowances[msg.sender][_spender] = 0;
555     } else {
556         allowances[msg.sender][_spender] = oldValue - _subtractedValue;
557     }
558     emit Approval(msg.sender, _spender, allowances[msg.sender][_spender]);
559     return true;
560 }
```

The code meets the specification

Detail for Request 83: decreaseApproval case 2

15, Jul 2018

Posted by CTK report generatc

110.6ms

Line 545-550 in File contentos.sol

```
545 /*@CTK "decreaseApproval case 2"
546   @pre stopped == false
547   @pre allowances[msg.sender][_spender] >= _subtractedValue
548   @post __post.allowances[msg.sender][_spender] == allowances[msg.sender][_spender
549   @post __return == true
550 */
```

Line 551-560 in File contentos.sol

```
551 function decreaseApproval(address _spender, uint _subtractedValue) stoppable publi
```

```
552     uint oldValue = allowances[msg.sender][_spender];
553     if (_subtractedValue > oldValue) {
554         allowances[msg.sender][_spender] = 0;
555     } else {
556         allowances[msg.sender][_spender] = oldValue - _subtractedValue;
557     }
558     emit Approval(msg.sender, _spender, allowances[msg.sender][_spender]);
559     return true;
560 }
```

The code meets the specification

Detail for Request 84: Buffer overflow / array index out of bound would never happen.

15, Jul 2018

Posted by CTK report generatc

92.5ms

Line 230 in File contentos.sol

```
230 //@CTK NO_BUF_OVERFLOW
```

Line 247-264 in File contentos.sol

```
247 function _transfer(address _from, address _to, uint _value) whenNotFreezed(_from)
248     // Prevent transfer to 0x0 address. Use burn() instead
249     require(_to != 0x0);
250     // Check if the sender has enough
251     require(balances[_from] >= _value);
252     // Check for overflows
253     require(balances[_to] + _value > balances[_to]);
254     // Save this for an assertion in the future
255     uint previousBalances = balances[_from] + balances[_to];
256     // Subtract from the sender
257     balances[_from] -= _value;
258     // Add the same to the recipient
259     balances[_to] += _value;
260     emit Transfer(_from, _to, _value);
261     // Asserts are used to use static analysis to find bugs in your code. They shc
262     assert(balances[_from] + balances[_to] == previousBalances);
263     return true;
264 }
```

The code meets the specification

Detail for Request 85: Method will not encounter an assertion failure.

15, Jul 2018

Posted by CTK report generatc

573.7ms

Line 231 in File contentos.sol

```
231 //@CTK NO_ASF
```

Line 247-264 in File contentos.sol

```
247 function _transfer(address _from, address _to, uint _value) whenNotFreezed(_from)
248     // Prevent transfer to 0x0 address. Use burn() instead
249     require(_to != 0x0);
250     // Check if the sender has enough
251     require(balances[_from] >= _value);
252     // Check for overflows
253     require(balances[_to] + _value > balances[_to]);
254     // Save this for an assertion in the future
255     uint previousBalances = balances[_from] + balances[_to];
256     // Subtract from the sender
257     balances[_from] -= _value;
258     // Add the same to the recipient
259     balances[_to] += _value;
260     emit Transfer(_from, _to, _value);
261     // Asserts are used to use static analysis to find bugs in your code. They shc
262     assert(balances[_from] + balances[_to] == previousBalances);
263     return true;
264 }
```

The code meets the specification

Detail for Request 86: transfer

15, Jul 2018

Posted by CTK report generatc

834.7ms

Line 232-245 in File contentos.sol

```
232 /*@CTK "transfer"
233   @tag assume_completion
234   @pre _to != 0x0
235   @pre balances[_from] >= _value
236   @pre balances[_to] + _value > balances[_to]
237   @pre balances[_to] + balances[_from] >= balances[_to]
238   @pre _freezeList[_from] == false
```

```
  @post !__has_overflow
  @post _from != _to -> __post.balances[_to] == balances[_to] + _value
  @post _from != _to -> __post.balances[_from] == balances[_from] - _value
  @post _from == _to -> __post.balances[_to] == balances[_to]
  @post _from == _to -> __post.balances[_from] == balances[_from]
  @post __return == true
*/
```

Line 247-264 in File contentos.sol

```
function _transfer(address _from, address _to, uint _value) whenNotFreezed(_from)
    // Prevent transfer to 0x0 address. Use burn() instead
    require(_to != 0x0);
    // Check if the sender has enough
    require(balances[_from] >= _value);
    // Check for overflows
    require(balances[_to] + _value > balances[_to]);
    // Save this for an assertion in the future
    uint previousBalances = balances[_from] + balances[_to];
    // Subtract from the sender
    balances[_from] -= _value;
    // Add the same to the recipient
    balances[_to] += _value;
    emit Transfer(_from, _to, _value);
    // Asserts are used to use static analysis to find bugs in your code. They shc
    assert(balances[_from] + balances[_to] == previousBalances);
    return true;
}
```

The code meets the specification

Detail for Request 87: If method completes, integer overflow would not happen.

15, Jul 2018

Posted by CTK report generatc

0.8ms

Line 217 in File contentos.sol

```
//@CTK NO_OVERFLOW
```

Line 223-225 in File contentos.sol

```
function allowance(address _owner, address _spender) public view returns (uint256)
    return allowances[_owner][_spender];
}
```

The code meets the specification

Detail for Request 88: Buffer overflow / array index out of bound would never happen.

15, Jul 2018

Posted by CTK report generatc

0.7ms

Line 218 in File contentos.sol

```
218 //@CTK NO_BUF_OVERFLOW
```

Line 223-225 in File contentos.sol

```
223 function allowance(address _owner, address _spender) public view returns (uint256)
224     return allowances[_owner][_spender];
225 }
```

The code meets the specification

Detail for Request 89: Method will not encounter an assertion failure.

15, Jul 2018

Posted by CTK report generatc

0.8ms

Line 219 in File contentos.sol

```
219 //@CTK NO_ASF
```

Line 223-225 in File contentos.sol

```
223 function allowance(address _owner, address _spender) public view returns (uint256)
224     return allowances[_owner][_spender];
225 }
```

The code meets the specification

Detail for Request 90: allowance

15, Jul 2018

Posted by CTK report generatc

1.6ms

Line 220-222 in File contentos.sol

```
220 /*@CTK "allowance"
221   @post __return == allowances[_owner][_spender]
222 */
```

Line 223-225 in File contentos.sol

```
223 function allowance(address _owner, address _spender) public view returns (uint256)
224     return allowances[_owner][_spender];
225 }
```

The code meets the specification

Detail for Request 91: Buffer overflow / array index out of bound would never happen.

15, Jul 2018

Posted by CTK report generatc

0.8ms

Line 189 in File contentos.sol

```
189 //@CTK NO_BUF_OVERFLOW
```

Line 198-205 in File contentos.sol

```
198 constructor(uint256 _initialSupply, string _tokenName, string _tokenSymbol, uint8
199     name = _tokenName;                                   // Set the name for displ
200     symbol = _tokenSymbol;                               // Set the symbol for dis
201     decimals = _decimals;
202     //owner = msg.sender;
203     totalSupply = _initialSupply * 10 ** uint256(decimals);  // Update total suppl
204     balances[owner] = totalSupply;                    // Give the creator all initial
205 }
```

The code meets the specification

Detail for Request 92: Method will not encounter an assertion failure.

15, Jul 2018

Posted by CTK report generatc

0.7ms

Line 190 in File contentos.sol

```
190 //@CTK NO_ASF
```

Line 198-205 in File contentos.sol

```
198 constructor(uint256 _initialSupply, string _tokenName, string _tokenSymbol, uint8
199     name = _tokenName;                                    // Set the name for displ
200     symbol = _tokenSymbol;                                // Set the symbol for dis
201     decimals = _decimals;
202     //owner = msg.sender;
203     totalSupply = _initialSupply * 10 ** uint256(decimals);  // Update total suppl
204     balances[owner] = totalSupply;                // Give the creator all initial
205 }
```

The code meets the specification

Detail for Request 93: constructor

15, Jul 2018

Posted by CTK report generatc

10.7ms

Line 191-197 in File contentos.sol

```
191     /*@CTK "constructor"
192       @post __post.name == _tokenName
193       @post __post.symbol == _tokenSymbol
194       @post __post.decimals == _decimals
195 //      @post __post.totalSupply == _initialSupply * 10 ** decimals
196       @post __post.balances[owner] == __post.totalSupply
197     */
```

Line 198-205 in File contentos.sol

```
198 constructor(uint256 _initialSupply, string _tokenName, string _tokenSymbol, uint8
199     name = _tokenName;                                    // Set the name for displ
200     symbol = _tokenSymbol;                                // Set the symbol for dis
```

```
201     decimals = _decimals;
202     //owner = msg.sender;
203     totalSupply = _initialSupply * 10 ** uint256(decimals);  // Update total suppl
204     balances[owner] = totalSupply;                // Give the creator all initial
205 }
```

The code meets the specification

Detail for Request 94: If method completes, integer overflow would not happen.

15, Jul 2018

Posted by CTK report generatc

1.3ms

Line 565 in File contentos.sol

```
565 //@CTK NO_OVERFLOW
```

Line 575-576 in File contentos.sol

```
575 constructor() COSTokenBase(10000000000, "Contentos", "COS", 18) public {
576 }
```

The code meets the specification

Detail for Request 95: Buffer overflow / array index out of bound would never happen.

15, Jul 2018

Posted by CTK report generatc

0.7ms

Line 566 in File contentos.sol

```
566 //@CTK NO_BUF_OVERFLOW
```

Line 575-576 in File contentos.sol

```
575 constructor() COSTokenBase(10000000000, "Contentos", "COS", 18) public {
576 }
```

The code meets the specification

Detail for Request 96: Method will not encounter an assertion failure.

15, Jul 2018

Posted by CTK report generatc

0.7ms

Line 567 in File contentos.sol

```
567 //@CTK NO_ASF
```

Line 575-576 in File contentos.sol

```
575 constructor() COSTokenBase(10000000000, "Contentos", "COS", 18) public {
576 }
```

The code meets the specification

Detail for Request 97: constructor

15, Jul 2018

Posted by CTK report generatc

6.7ms

Line 568-574 in File contentos.sol

```
568 /*@CTK "constructor"
569   @post __post.name == "Contentos"
570   @post __post.symbol == "COS"
571   @post __post.decimals == 18
572   @post __post.totalSupply == 1e28
573   @post __post.balances[owner] == 1e28
574 */
```

Line 575-576 in File contentos.sol

```
575 constructor() COSTokenBase(10000000000, "Contentos", "COS", 18) public {
576 }
```

The code meets the specification