

SOLIDITY COMPILER

COMPILER +

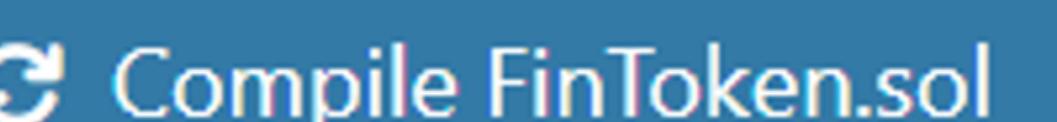
0.4.21+commit.dfe3193c

Include nightly builds

Auto compile

Hide warnings

Advanced Configurations >

 Compile FinToken.sol

Compile and Run script

FinToken.sol 1 X

```
* @dev Based on code by FirstBlood: https://github.com/Firstbloodio/token/blob/master/smart\_contract/FirstBloodToken.sol
*/
contract StandardToken is ERC20, BasicToken {

    mapping (address => mapping (address => uint256)) internal allowed;

    /**
     * @dev Transfer tokens from one address to another
     * @param _from address The address which you want to send tokens from
     * @param _to address The address which you want to transfer to
     * @param _value uint256 the amount of tokens to be transferred
     */
    function transferFrom(address _from, address _to, uint256 _value) public returns (bool) {
        require(_to != address(0));
        require(_value <= balances[_from]);
        require(_value <= allowed[_from][msg.sender]);

        balances[_from] = balances[_from].sub(_value);
        balances[_to] = balances[_to].add(_value);
        allowed[_from][msg.sender] = allowed[_from][msg.sender].sub(_value);
        emit Transfer(_from, _to, _value);
        return true;
    }

    /**
     * @dev Approve the passed address to spend the specified amount of tokens on behalf of msg.sender.
     *
     * Beware that changing an allowance with this method brings the risk that someone may use both the old
     * and the new allowance by unfortunate transaction ordering. One possible solution to mitigate this
     * race condition is to first reduce the spender's allowance to 0 and set the desired value afterwards:
     * https://github.com/ethereum/ETPs/issues/20#issuecomment-263524729
     */
}
```

0  Listen on all transactions C

Compile .sol File

Target Function for Debug