

Excerpt from a sample smart contract code for a mining royalty token

The unaudited excerpt below is based on Solidity, a contract-oriented programming language for writing smart contracts. It is used for implementing smart contracts on various blockchain platforms, including Ethereum. Specific royalty features, such as the exact commercial content of the royalty stream, have not been included in this code. This fragment represents the simplest form of token that can be issued on the Ethereum network.

```
pragma solidity ^0.4.18;
contract ERC20Interface {
    function totalSupply() public constant returns (uint);
    function balanceOf(address tokenOwner) public constant returns (uint balance);
    function allowance(address tokenOwner, address spender) public constant returns (uint remaining);
}
contract MiningRoyaltyToken is ERC20Interface, Owned, SafeMath {
    string public symbol;
    string public name;
    uint8 public decimals;
    uint public _totalSupply;
    uint public startDate;
    uint public bonusEnds;
    uint public endDate;
    mapping(address => uint) balances;
    mapping(address => mapping(address => uint)) allowed;
    function MiningRoyaltyToken() public {
        symbol = "MRT";
        name = "MiningRoyalty Token";
        decimals = 18;
        bonusEnds = now + 1 weeks;
        endDate = now + 7 weeks;
    }
    function totalSupply() public constant returns (uint) {
        return _totalSupply - balances[address(0)];
    }
    function balanceOf(address tokenOwner) public constant returns (uint balance) {
        return balances[tokenOwner];
    }
}
```