Scribble Functions

How to write User defined functions User-defined scribble functions are useful to avoid repeating similar expressions in multiple annotations. They are defined in the docstring above a smart contract using annotations structured as shown below:

```

Copy /// #define {:msg "Description of your function"} /// funName(type1 arg1, type2 arg,...) returnType ;

```

The above annotation defines a new user-defined function with namefunName that takes several arguments (arg1 of typetype1 ,arg2 of typetype2 , etc.), and has as its body a single valid Scribble expression (which is also its return value). That expression must be of typereturnType .

Consider the following example. We can see that the operations(x + 10) and(x + 10) * delta are repeated multiple times in annotations.

```

Copy contract ExampleContract { uint delta; function set_delta(uint _delta) public { delta = _delta; } /// #if_succeeds {:msg "P0"} y == (x + 10) * delta + 15; function example(uint x, uint y) public returns (uint result) { ... }

/// #if_succeeds {:msg "P1"} y == (x + 10) * delta + x + 10; function example2(uint x, uint y) public returns (uint result) { ... }

}

```

We can avoid those repetitions by adding user defined functions for(x+10) and(x+10) * delta above the smart contract. These functions can then be called by annotations inside the contract.

```

Copy /// #define add10(uint x) uint = x + 10; /// #define add10_times_delta(uint y) uint = add10(y) * delta; contract ExampleContract { uint delta; function set_delta(uint _delta) public { delta = _delta; } /// #if_succeeds {:msg "P0"} y == add10_times_delta(x) + 15; function example(uint x, uint y) public returns (uint result) { ... }

/// #if_succeeds {:msg "P1"} y == add10_times_delta(x) + add10(x); function example2(uint x, uint y) public returns (uint result) { ... }

}

```

In summary there are several important functions to remember about scribble user-defined functions:

1. They are defined in the docstring above a smart contract
2. They can be used by any annotation inside that contract, or in all inheriting contracts
3. Scribble user defined functions must beview
4. orpure
5. functions. I.e. they cannot modify any state, but can still access the state variables as shown in the above example.
6. The body of a user-defined function consists of a single valid Scribble expression, which is also the return value of the function
7.

User defined functions can also be used by all inheriting contracts.

Previous Contract Invariants Next Assert Annotations Last updated2 years ago

On this page Was this helpful?