Annotations

In Scribble you can formalise properties by writing annotations for contracts, state variables and functions. In addition you can add auxilary annotations that define helper functions to be used in properties, or that give hints to the backend tools that try to check the properties you wrote.

All annotations appear in docstrings, and begin with the pound sign# followed by a specific keyword. There are several different kinds of annotations at your disposal:

If you place an annotation in a normal comment (// or //), instead of a docstring comment (/// or ///) it will be ignored! Note that if you forget the pound sign (#) your annotation will also be ignored! Scribble will issue a warning in most cases when this happens. Solidity recently added the@custom: natspec tag. As of Scribblev0.6.1 we also support annotations prefixed with@custom:scribble . Note that the pound sign before keywords is still requried! (i.e. you need to write@custom:scribble #if_succeeds ... instead of@custom:scribble if_succeeds ...).

Function Annotations

Function annotations are what you can use to describe how a function should behave. You'll commonly use them to ensure that the function has the right side-effects, and that the return values are correct.

Example

```

Copy /// #if_succeeds result > 0; function positiveNumber() external returns (uint) { ... }

```

Read more about function annotations here:

page Function Annotations

State Variable Annotations

State variable annotations allow you to describe properties that should hold for a particular variable whenver its updated. For example, you might specify that a variable can only change under particular circumstances.

Example

```

Copy contract Example { // #if_updated msg.sender == owner || owner == address(0); address owner; }

```

Read more about state variable annotations here:

page State Variable Annotations

Contract Annotations

Contract annotations allow you to describe properties that concern multiple variables and functions. The most common contract annotation will be an invariant, which is used to describe properties that hold before and after each transaction.

Example

```

Copy /// #invariant sum(balances) == totalSupply; contract Token { ... }

```

Read more about contract annotations here:

page Contract Invariants

Scribble Functions

Scribble functions are a utility feature which enables you to reduce code duplication between annotations. They also make annotations much easier to read!

Example

```

Copy /// #define add10(uint x) uint = x + 10; contract Example { /// #if_succeeds add10(1) == 11; function example() public {} }

```

Read more about Scribble functions here:

page Scribble Functions

Assert Annotations

Assert annotations allow users to insert an annotation in the middle of a function, before a specific statement. They can refer to any local variable in scope

Example

```

Copy contract Example { mapping(address=>uint) balances; function setBalances(address memory[] addrs, uint[] memory amounts) public { for (uint i = 0; i < addrs.length; i++) { /// #assert amounts[i] > 0; balances[addrs[i]] = amounts[i]; } } }

```

Read more about Scribble functions here:

page Assert Annotations

Macros

Macro annotations allow applying a whole set of pre-defined annotations to a contract. Macro annotations are defined in.yaml files and applied to a contract with the/// #macro name(var1, var2, ...). annotation.

Example

```

Copy /// #macro erc20(_balances); contract MyERC20Token { mapping(address=>uint256) _balances; .... }

```

Read more about Scribble macros here:

page Macros

Hints

Sometimes a user needs to provide guidance to the backend tools that are attempting to check the annotated properties. Scribble supports providing guidance with 2 types of annotations:

1. Limiting the types of input with a#require
2. annotations
3. Suggesting inputs for the tools to try with#try
4. annotations
5.

Read more about giving hints to the backend tools here:

page Hints

Previous Introduction Next Function Annotations Last updated2 years ago

On this page * Function Annotations * State Variable Annotations * Contract Annotations * Scribble Functions * Assert Annotations * Macros * Hints

Was this helpful?