

# CMTAT Test Framework

## Table of Contents

Guideline.....	2
How write a test ?.....	2
Checklist.....	4
PauseModule.....	4
MintModule.....	4
BurnModule.....	4
ValidationModule.....	5
BaseModule.....	5
Proxy (test Z).....	5
Test list.....	6
Test Z – Proxy.....	6
Test Z1.....	6
Test Z2.....	6
Test Z3.....	7
Test A - PauseModule.....	8
Test B - MintModule.....	10
Test C - BurnModule.....	11
Test D - ValidationModule.....	12

## Guideline

It is important that the tests can easily be improved and understood by others.

For each test file, the list of tests must be present.

## How write a test ?

The test must follow the pattern AAA for the documentation and the structure.

First, read this excellent document by [Microsoft](#).

Here a little resume :

Term	Definition
Arrange	Arrange your objects, create and set them up as necessary.
Arrange - Assert	Assertion to check your arrange
Act	The tested function
Assert	All check to verify the result obtained by the call of the function(s) in the Act part.

### New test file

- Create a new tab with a new Id [A,B, C.....]
- Create a new tab in the section checklist

### New test

For each new test : add an entry after the previous ones in the corresponding table

Example : you create a new test called *testCanTransferIsTrue* in the file RuleWhitelist.t.sol. You add then an entry in the corresponding table. After that, add the test in the checklist too.

Below is an example of an entry in the table

id	Test function	Truffle/ Foundry	Target function	Expected result	Event check	Truffle Actual result	Foundry Actual result	conclusion	Improvement
[...previous test]									
25	testCanTransferIs True	[both, Foundry, Truffle]	The tested function	What is the result supposed to be returned by the function ???	[yes, no, -] “no” means “events are not checked” “-” means “there are no events to check”	Test with Truffle [As expected] or[Not as expected + the result]	Test with Foundry [As expected] or[Not as expected + the result]	[Ok, Not Ok]	Possible improvement for the test

## Checklist

The checklist allows you to quickly check that all the functions are tested as well as to find the corresponding test

### PauseModule

File : **PauseModule.sol**

Functions	Test id
pause	A1, A2, A3, A7, A8
unpause	A4, A5, A6

### MintModule

File : **MintModule.sol**

Functions	Test id
mint	B1, B2, B3

### BurnModule

File : **BurnModule.sol**

Functions	Test id
burnFrom	C1,C2,C3,C4

## ValidationModule

File : ValidationModule.sol

Functions	Test id
setRuleEngine	D1, D2
detectTransferRestriction	D3, D5
messageForTransferRestriction	D4, D6
transfer	D7, D8
mint	-

## BaseModule

File: BaseModule.sol

## Proxy (test Z)

Functions	Test id
Kill	Z1/1, Z2/1, Z2/2, Z2/3
UpgradeProxy (Truffle Plugin function)	Z3/1

## Test list

### Test Z – Proxy

#### Kill Implementation

We use a different version of the CMTAT where we have removed the check of access control on the kill function

The goal is to verify if the modifier *onlyDelegateCall* works as intended

#### Test Z1

Target File : CMTAT.sol

Test files: KillImplementation.test.js (Truffle)

id	Test function	Truffle/ Foundry	Target function	Expected result	Event check	Actual result	Conclusion	Improvement
1	testCannotKillTheImplementationContract	Truffle	kill	The contract is not killed	Yes	As expected	Ok	

#### Test Z2

Target File : CMTAT.sol

Test files: Proxy.test.js (Truffle)

id	Test function	Truffle /	Target function	Expected result	Event	Actual result	Concl	Improvement
----	---------------	--------------	-----------------	-----------------	-------	---------------	-------	-------------

		Foundry			check		usion	
1	testCannotBeTakenControlByAttacker1	Truffle	kill	-The attacker can not take control of the implementation contract.  -It can not execute the function kill, an error is generated.	-	As expected	Ok	
2	testCannotBeTakenControlByAttacker2	Truffle	kill	Same result than testCannotBeTakenControlByAttacker1	-	As expected	Ok	
3	testCannotKillTheImplementationContractByAdmin	Truffle	kill	The admin can not execute the function kill, an error is generated.	-	As expected	Ok	

### Test Z3

Target File : CMTAT.sol

Test files: UpgradeProxy.test.js (Truffle)

id	Test function	Truffle / Found	Target function	Expected result	Event check	Actual result	Conclusion	Improvement
----	---------------	-----------------	-----------------	-----------------	-------------	---------------	------------	-------------

		ry						
1	testKeepStorageFor Tokens	Truffle	upgradeProxy	The proxy is upgraded with the new implementation and keeps its storage for the tokens balance.	-	As expected	Ok	

## Test A - PauseModule

Target File: PauseModule.sol

Target File: MintModule.sol

Test files: MintModuleCommon.js (Truffle), MintModule.t.sol(Foundry)

id	Test function	Truffle / Foundry	Target function	Expected result	Event check	Actual result	conclusion	Improvement
1	testCanBePausedBy Admin	Both	pause	The contract is in pause	Yes	As expected	Ok	
2	testCanBePausedBy ANewPauser	Both	pause	The contract is in pause	Yes	As expected	Ok	



3	testCannotBePausedByNonPauser	both	pause	Revert because the sender has not the right role.	-	As expected	Ok	
4	testCanBeUnpausedByAdmin	both	unpause	A contract in pause can get out from this state with a call to the unpause function by the admin	Yes	As expected	Ok	
5	TestCanBeUnpausedByANewPauser	both	unpause	A contract in pause can get out from this state with a call to the unpause function by an address with the right role (PAUSER_ROLE)	Yes	As expected	OK	
6	testCannotBeUnpausedByNonPauser	both	unpause	Revert because the sender has not the right role.	-	As expected	Ok	
7	testCannotTransferTokenWhenPaused_A	both	pause	The transfer is reverted because the contract is in pause	-	As expected	Ok	
8	testCannotTransferTokenWhenPaused_B	both	pause	The transfer is reverted because the contract is in pause	-	As expected	Ok	

## Test B - MintModule

Target File : MintModule.sol

Test files: MintModuleCommon.js (Truffle), MintModule.t.sol(Foundry)

id	Test function	Truffle/ Foundry	Target function	Expected result	Event check	Truffle Actual result	Foundry Actual result	conclusion	Improvement
1	testCanBeMintedByAdmin	Both	mint	The tokens are minted	Yes	As expected	As expected	Ok	
2	testCanBeMintedByANewMinter	Both	mint	The tokens are minted	Yes	As expected	As expected	Ok	
3	testCannotIssuingByNonMinter	Both	mint	Revert because the sender has not the right role.	-	As expected	As expected	OK	

## Test C - BurnModule

Target File : **BurnModule.sol**

id	Test function	Truffle / Foundry	Target function	Expected result	Event check	Truffle Actual result	Foundry Actual result	conclusion	Improvement
1	testCanBeBurntByAdminWithAllowance	Both	burnFrom	The tokens are burn	Yes		As expected	Ok	
2	testCanBeBurntByBurnerRole	Both	burnFrom	The tokens are burn	Yes		As expected	Ok	
3	testCannotBeBurntWithoutAllowance	Both	burnFrom	Revert because the sender has not sufficient allowance on the tokens	-		As expected	Ok	
4	testCannotBeBurntWithoutBurnerRole	Both	burnFrom	Revert because the sender has not the right role	-		As expected	Ok	

## Test D - ValidationModule

Target File : **ValidationModule.sol**

id	Test function	Truffle / Foundry	Target function	Expected result	Event check	Truffle Actual result	Foundry Actual result	conclusion	Improvement
1	testCanBeSetByAdmin	both	setRuleEngine	The RuleEngine is set	Yes	As expected	As expected	Ok	
2	testCannotBeSetByNonAdmin	both	setRuleEngine	The transaction is reverted	-	As expected	As expected	Ok	
3	testCanDetectTransferRestrictionValidTransfer	both	detectTransferRestriction	The returned code corresponds to that of a valid transfer	-	As expected	As expected	Ok	
4	testCanReturnMessageValidTransfer	both	messageForTransferRestriction	The returned message corresponds to that of a valid transfer	-	As expected	As expected	Ok	
5	testCanDetectTransferRestrictionWithAmount	both	detectTransferRestriction	The returned code corresponds to that of an invalid transfer in reason of excessive amount	-	As expected	As expected	Ok	

	ntTooHigh								
6	testCanReturn MessageWithA mountTooHigh	both	messageForTransfer Restriction	The returned message corresponds to that of a invalid transfer in reason of excessive amount	-		As expected	Ok	
7	testCanTransf erAllowedByR ule	both	transfer	The transfer is performed	No		As expected	Ok	
8	testCannotTra nsferIfNotAllow edByRule	both	transfer	The transfer is not performed, the transaction is reverted.	No		As expected	Ok	