# ETHALER

## Contributors

| Name | Organization |
|------|--------------|
| **Vipin Bharathan** | dlt.nyc |
| **Mani Pillai** | SwapsHub |

## Taxonomy Formula: tF{d,t,p,c,SC}

## Token Specification Summary

## Token Classification

| Template Type: | SingleToken | This token has no sub or child tokens. |
|----------------|-------------|----------------------------------------|
| **Token Type:** | Fungible | Tokens have interchangeable value with one another, where any quantity of them has the same value as another equal quantity if they are in the same class or series. |
| **Token Unit:** | Fractional | This token can be sub-divided or split into smaller units or parts based on a certain number of decimal places. |
| **Value Type:** | Intrinsic | This token is purely a digital token represents value directly, it represents no external physical form and cannot be a receipt or title for a material item or property. |
| **Representation Type:** | Common | This token is simply represented as a balance or quantity attributed to an owner address where all the balances are recorded on the same balance sheet, like a bank account. All instances can easily share common properties and locating them is simple. |
| **Supply:** | Infinite | Infinite supply indicates that tokens in the class can be created and removed with no cap and also potentially reflect negative supply for certain business cases. |

*This is a Token with Variable Supply Fungible where an initial supply can set at creation and then supply can be added and removed from the total based on need. It is fractional by setting the Decimals property on the dividable*

*behavior. A token instance can be burned or minted. Before executing transfer, burn or mint operation check if they are within the compliance regulations. Pausable for possible freezing of movement and all other operations because of discovered bugs or upgrade.*

## Example

Enables the issuance of regulated electronic money by the central bank (mintable and burnable only by Central bank) and its practical usage in real financial applications.

## EThaler is:

- Divisible
- Transferable
- Pausable
- Compliant
- Burnable
- Roles
- Mintable

# EThaler Details

## Fractional Fungible

| Type: | Base |
|---|---|
| Name: | Fractional Fungible |
| Id: | 89ca6daf-5585-469e-abd1-19bc44e7a012 |
| Visual: | &tau;<sub>F</sub>{<i>d</i>} |
| Tooling: | tF{d} |
| Version: | 1.0 |

## Definition

*Fractional Fungible tokens have interchangeable value with each other, where any owned sum of them from a class has the same value as another owned sum from the same class. Similar to physical cash money, a crypto currency is an example of a fungible token that is divisible.*

## Example

Fiat currency is the most widely understood example of a fractional fungible item. A fractional fungible is divisible, so you can 'make change'.

EThaler - ac49c603701963174e669577afa0112bbd0fe5d3fe29c06fc5a6a3cf794afad22

## Analogies

| Name | Description |
|------|-------------|
| **Physical Money or Cash** | Cash, or fiat money, is freely accepted between parties and can have varying denominations. Money has a face value, on a coin or bill, and can be summed together to represent higher value. It can be divided, making change, and consolidated from many smaller denominations to larger ones and still have the same value. |
| **General Admission Movie Ticket** | Purchasing a general admission ticket to a movie only allows for you to have a seat, but seat that you actually get depends on factors like when you arrive. |

## Dependencies

| Artifact Type | Symbol | Description |
|---------------|--------|-------------|
| **Base** | t | Base Token Definition |

## Incompatible With

| Artifact Type | Symbol | Id |
|---------------|--------|-----|
| **Behavior** | ~d | d5807a8e-879b-4885-95fa-f09ba2a22172 |

## Influenced By

| Description | Symbol | Applies To |
|-------------|--------|------------|

## Artifact Files

| Content Type | File Name | File Content |
|--------------|-----------|--------------|
| **Control** | fractional-fungible.proto | |
| **Uml** | fractional-fungible.md | |

## Code Map

| Map Type | Name | Platform | Location |
|----------|------|----------|----------|

EThaler - ac49c603701963174e669577afa0112bbd0fe5d3fe29c06fc5a6a3cf794afad23

## Implementation Map

| Map Type | Name | Platform | Location |
|----------|------|----------|----------|
| | | | |

## Resource Map

| Map Type | Name | Location | Description |
|----------|------|----------|-------------|
| | | | |

## Base Details

| Token Name: | |
|-------------|---|
| **Token Type:** | Fungible |
| **Representation Type:** | Common |
| **Value Type:** | Intrinsic |
| **Token Unit:** | Fractional |
| **Symbol:** | |
| **Owner:** | |
| **Quantity:** | 0 |
| **Decimals:** | 0 |
| **Constructor Name:** | Constructor |

## Behaviors

## Divisible

| Type: | Behavior |
|-------|----------|
| **Name:** | Divisible |
| **Id:** | 6e3501dc-5800-4c71-b59e-ad11418a998c |
| **Visual:** | <i>d</i> |
| **Tooling:** | d |
| **Version:** | 1.0 |

# Definition

*An ability for the token to be divided from a single whole token into fractions, which are represented as decimal places. Any value greater than 0 will indicate how many fractions are possible where the smallest fraction is also the smallest ownable unit of the token.*

# Example

Divisible is common for crypto-currencies or tokens of fiat currency.  For example, the US Dollar is divisible to 2 decimal places, where a value like .42 is possible.  Bitcoin, is divisible up to 8 decimal places.

# Analogies

| Name | Description |
|------|-------------|
| **Analogy 1** | divisible analogy 1 description |

# Dependencies

| Artifact Type | Symbol | Description |
|---------------|--------|-------------|

# Incompatible With

| Artifact Type | Symbol | Id |
|---------------|--------|-----|
| **Behavior** | ~d | d5807a8e-879b-4885-95fa-f09ba2a22172 |

# Influenced By

| Description | Symbol | Applies To |
|-------------|--------|------------|

# Artifact Files

| Content Type | File Name | File Content |
|--------------|-----------|--------------|
| **Control** | divisible.proto | |
| **Uml** | divisible.md | |

EThaler - ac49c603701963174e669577afa0112bbd0fe5d3fe29c06fc5a6a3cf794afad25

## Code Map

| Map Type | Name | Platform | Location |
|----------|------|----------|----------|

## Implementation Map

| Map Type | Name | Platform | Location |
|----------|------|----------|----------|

## Resource Map

| Map Type | Name | Location | Description |
|----------|------|----------|-------------|

## Specification Behavior

# Divisible

## Taxonomy Symbol: d

*An ability for the token to be divided from a single whole token into fractions, which are represented as decimal places. Any value greater than 0 will indicate how many fractions are possible where the smallest fraction is also the smallest ownable unit of the token.*

## Example

Divisible is common for crypto-currencies or tokens of fiat currency.  For example, the US Dollar is divisible to 2 decimal places, where a value like .42 is possible.  Bitcoin, is divisible up to 8 decimal places.

## Analogies

| Name | Description |
|------|-------------|
| **Analogy 1** | divisible analogy 1 description |

| Is External: | True |
|--------------|------|
| **Constructor:** | |

# Divisible responds to these Invocations

## Properties

## *Name: Decimals*

Value Description: Set to a number greater than Zero, allowing subdivision

Template Value: 4

## Invocations

## *GetDecimals*

Id: 01f7ef04-1215-45f1-b118-12b4a76db9ad

Description: Return the value

## Request

Control Message: GetDecimalsRequest

Description:

## *Parameters*

| Name | Value |
|------|-------|
|      |       |

## Response

Control Message: GetDecimalsResponse

Description: Return number of decimal places

## *Parameters*

| Name | Value |
|------|-------|
| **Decimals** | integer |

## *GetDecimals*

Id: 01f7ef04-1215-45f1-b118-12b4a76db9ad

Description: Return the value

## Request

Control Message: GetDecimalsRequest

Description:

*Parameters*

| Name | Value |
|------|-------|
|      |       |

## Response

Control Message: GetDecimalsResponse

Description: Return number of decimal places

*Parameters*

| Name | Value |
|------|-------|
| **Decimals** | integer |

# Properties

## Transferable

| Type: | Behavior |
|-------|----------|
| **Name:** | Transferable |
| **Id:** | af119e58-6d84-4ca6-9656-75e8d312f038 |
| **Visual:** | <i>t</i> |
| **Tooling:** | t |
| **Version:** | 1.0 |

## Definition

*Every token instance has an owner. The Transferable behavior provides the owner the ability to transfer the ownership to another party or account. This behavior is often inferred by other behaviors that might exist like Redeem, Sell, etc. This behavior is Delegable. If the token definition is Delegable, TransferFrom will be available.*

## Example

## Analogies

| Name | Description |
|------|-------------|
| **Analogy 1** | transferable analogy 1 description |

## Dependencies

| Artifact Type | Symbol | Description |
|---|---|---|

## Incompatible With

| Artifact Type | Symbol | Id |
|---|---|---|
| **Behavior** | ~t | a4fa4ca8-6afd-452b-91f5-7103b6fee5e5 |

## Influenced By

| Description | Symbol | Applies To |
|---|---|---|
| **If the token is Delegable, TransferFrom should be enabled.** | g | [ ] |
| **If Compliance is present, a CheckTransferAllowed request has to be made and verified before a Transfer request or a TransferFrom request.** | c | [ ] |
| **If issuable is present, an AcceptTokenRequest from the token issuer, in response to a RequestTokens, has to be made and verified before a Transfer request.** | i | [ ] |

## Artifact Files

| Content Type | File Name | File Content |
|---|---|---|
| **Control** | transferable.proto | |
| **Uml** | transferable.md | |

## Code Map

| Map Type | Name | Platform | Location |
|---|---|---|---|

## Implementation Map

| Map Type | Name | Platform | Location |
|---|---|---|---|

## Resource Map

| Map Type | Name | Location | Description |
|----------|------|----------|-------------|
|          |      |          |             |

## Specification Behavior

# Transferable

## Taxonomy Symbol: t

*Every token instance has an owner. The Transferable behavior provides the owner the ability to transfer the ownership to another party or account. This behavior is often inferred by other behaviors that might exist like Redeem, Sell, etc. This behavior is Delegable. If the token definition is Delegable, TransferFrom will be available.*

## Example

## Analogies

| Name | Description |
|------|-------------|
| **Analogy 1** | transferable analogy 1 description |

| Is External: | True |
|--------------|------|
| **Constructor:** | |

## Transferable responds to these Invocations

### *Transfer*

Id: 5d4b8f10-7857-4a2f-9b8c-d61e367a6bcc

Description: >A transfer request will invoke a transfer from the owner of the token to the party or account provided in the To field of the request. For fungible or dividable non-fungible tokens, this request may also include value in the Amount field of the request to transfer more than one token of the class in a single request.

## Request Message:

TransferRequest

Description: The request

| Name | Value |
|------|-------|
| **To** | AccountId to transfer ownership to. |
| **Quantity** | Number of tokens to transfer. |

## Response Message

TransferResponse

Description: The response

*Response Parameters*

| Name | Value |
|------|-------|
| **Confirmation** | A confirmation receipt or error may be returned to the owner based on the outcome of the transfer request. |

# *TransferFrom*

Id: 516b4e2f-4a14-4c4f-a6f2-1419d4af35c6

Description: >A transfer request will invoke a transfer from the owner of the token to the party or account provided in the To field of the request. For fungible or dividable non-fungible tokens, this request may also include value in the Amount field of the request to transfer more than one token of the class in a single request.

## Request Message:

TransferFromRequest

Description: The request

*Request Parameters*

| Name | Value |
|------|-------|
| **From** | AccountId to transfer ownership from. |
| **To** | AccountId to transfer ownership to. |
| **Quantity** | Number of tokens to transfer. |

## Response Message

TransferFromResponse

EThaler - ac49c603701963174e669577afa0112bbd0fe5d3fe29c06fc5a6a3cf794afad211

Description: The response

| Name | Value |
|------|-------|
| **Confirmation** | A confirmation receipt or error may be returned to the owner based on the outcome of the transfer from request. |

# Properties

## Pausable

| Type: | Behavior |
|-------|----------|
| **Name:** | Pausable |
| **Id:** | 0fb5abae-c99e-4f34-90cf-62b6f3351b74 |
| **Visual:** | <i>p</i> |
| **Tooling:** | p |
| **Version:** | 1.0 |

## Definition

*Pausable is an influencing behavior that can be applied to other behaviors in the Token. Pausable applies to a token class that implements this behavior; will halt trades and free all transfers, handy if there is a bug found in the token implementation.*

## Example

There may be a run or a crash in the market that may require the halting of trades for this token. This is like the big red button.

## Analogies

| Name | Description |
|------|-------------|
| **Bug in Code** | You may discover a bug in your token implementation that requires you to halt the trading until you can fix the code. |

## Dependencies

| Artifact Type | Symbol | Description |
|---|---|---|
| | | |

## Incompatible With

| Artifact Type | Symbol | Id |
|---|---|---|
| | | |

## Influenced By

| Description | Symbol | Applies To |
|---|---|---|
| **Roles can influence who or what role can pause and resume a token.** | r | [ ] |

## Artifact Files

| Content Type | File Name | File Content |
|---|---|---|
| **Control** | pausable.proto | |
| **Uml** | pausable.md | |

## Code Map

| Map Type | Name | Platform | Location |
|---|---|---|---|
| **SourceCode** | Open Zeppelin ERC-20 | EthereumSolidity | https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/token/ERC20/ERC20Pausable.sol |
| **SourceCode** | Open Zeppelin ERC-751 | EthereumSolidity | https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/token/ERC721/ERC721Pausable.sol |

## Implementation Map

| Map Type | Name | Platform | Location |
|---|---|---|---|
| | | | |

## Resource Map

| Map Type | Name | Location | Description |
|----------|------|----------|-------------|
|          |      |          |             |

## Specification Behavior

# Pausable

## Taxonomy Symbol: p

*Pausable is an influencing behavior that can be applied to other behaviors in the Token. Pausable applies to a token class that implements this behavior; will halt trades and free all transfers, handy if there is a bug found in the token implementation.*

## Example

There may be a run or a crash in the market that may require the halting of trades for this token. This is like the big red button.

## Analogies

| Name | Description |
|------|-------------|
| **Bug in Code** | You may discover a bug in your token implementation that requires you to halt the trading until you can fix the code. |

| Is External: | True |
|--------------|------|
| **Constructor:** | |

## Pausable responds to these Invocations

### *Pause*

Id: 2e0fd8e5-2090-4c62-b094-232c32a78022

Description: A Request to pause behavior invocations that Pausable applies to.

### Request Message:

PauseRequest

Description: The request

| Name | Value |
|------|-------|
|      |       |

## Response Message

PauseResponse

Description: The response

*Response Parameters*

| Name | Value |
|------|-------|
| **Confirmation** | A confirmation receipt or denial be returned to the pause requestor. |

## *Resume*

Id: 6d5df99d-2f5e-4c7a-aea4-d2d54176abfd

Description: Resume normal operations.

## Request Message:

ResumeRequest

Description: The request

*Request Parameters*

| Name | Value |
|------|-------|
|      |       |

## Response Message

ResumeResponse

Description: The response

*Response Parameters*

| Name | Value |
|------|-------|
| **Confirmation** | A confirmation response from the owner approving the an allowance request, indicating a allowance quantity the requestor has the option to invoke the Delegable behaviors on the token(s). |

# Properties

## Compliant

| Type: | Behavior |
|---|---|
| Name: | Compliant |
| Id: | 03dd1c48-dfdb-4ec1-86c8-69c3abac76b7 |
| Visual: | <i>c</i> |
| Tooling: | c |
| Version: | 1.0 |

## Definition

*A regulated token needs to comply with several legal requirements, especially KYC and AML. If the necessary checks have to be made off-chain the token transfer becomes centralized. Further the transfer in this case takes longer to complete as it can not be done in one transaction, but requires a second confirmation step. A compliant token fulfills all legal requirements on-chain without interaction from an off-chain entity.*

## Example

When doing a bank transfer the transaction is checked by the involved banks according to legal requirements.

## Analogies

| Name | Description |
|---|---|
| | |

## Dependencies

| Artifact Type | Symbol | Description |
|---|---|---|
| | | |

## Incompatible With

| Artifact Type | Symbol | Id |
|---|---|---|
| | | |

## Influenced By

| Description | Symbol | Applies To |
|---|---|---|
| | | |

EThaler - ac49c603701963174e669577afa0112bbd0fe5d3fe29c06fc5a6a3cf794afad216

## Artifact Files

| Content Type | File Name | File Content |
|---|---|---|
| **Control** | compliant.proto | |
| **Uml** | compliant.md | |

## Code Map

| Map Type | Name | Platform | Location |
|---|---|---|---|

## Implementation Map

| Map Type | Name | Platform | Location |
|---|---|---|---|

## Resource Map

| Map Type | Name | Location | Description |
|---|---|---|---|

## Specification Behavior

# Compliant

## Taxonomy Symbol: c

*A regulated token needs to comply with several legal requirements, especially KYC and AML. If the necessary checks have to be made off-chain the token transfer becomes centralized. Further the transfer in this case takes longer to complete as it can not be done in one transaction, but requires a second confirmation step. A compliant token fulfills all legal requirements on-chain without interaction from an off-chain entity.*

## Example

When doing a bank transfer the transaction is checked by the involved banks according to legal requirements.

| Is External: | True |
|---|---|

**Constructor:**

# Compliant responds to these Invocations

## *CheckTransferAllowed*

Id: 3f591127-0508-445b-b449-4adc3d8d90e9

Description: Checks if the transfer request is allowed to be executed with the given parameters.

## Request Message:

CheckTransferAllowedRequest

Description: The request

### *Request Parameters*

| Name | Value |
|------|-------|
| **From** | AccountId to transfer ownership from. |
| **To** | AccountId to transfer ownership to. |
| **Quantity** | Number of tokens to transfer. |

## Response Message

CheckTransferAllowedResponse

Description: The response

### *Response Parameters*

| Name | Value |
|------|-------|
| **Result** | A boolean value whereas true means the transfer is allowed and false means it is not. |

## *CheckMintAllowed*

Id: 0323b374-71af-48f6-93ff-2a63366267db

Description: Checks if the mint request is allowed to be executed with the given parameters.

## Request Message:

CheckMintAllowedRequest

Description: The request

EThaler - ac49c603701963174e669577afa0112bbd0fe5d3fe29c06fc5a6a3cf794afad218

| Name | Value |
|------|-------|
| **ToAccount** | Account Id to mint the tokens to. |
| **Quantity** | Number of tokens to transfer. |

## Response Message

CheckMintAllowedResponse

Description: The response

*Response Parameters*

| Name | Value |
|------|-------|
| **Result** | A boolean value whereas true means the minting request is allowed and false means it is not. |

## *CheckBurnAllowed*

Id: 8edffc4d-d14e-4a98-8c96-338835d5534c

Description: Checks if the burn request is allowed to be executed with the given parameters.

## Request Message:

CheckBurnAllowedRequest

Description: The request

*Request Parameters*

| Name | Value |
|------|-------|
| **From** | AccountId to transfer ownership from. |
| **Quantity** | Number of tokens to transfer. |

## Response Message

CheckBurnAllowedResponse

Description: The response

*Response Parameters*

| Name | Value |
|------|-------|
|  |  |

EThaler - ac49c603701963174e669577afa0112bbd0fe5d3fe29c06fc5a6a3cf794afad219

| | |
|---|---|
| **Result** | A boolean value whereas true means the burn request is allowed and false means it is not. |

## Properties

## Burnable

| Type: | Behavior |
|---|---|
| **Name:** | Burnable |
| **Id:** | 803297a1-c0f9-4898-9d44-29c9d41cca97 |
| **Visual:** | <i>b</i> |
| **Tooling:** | b |
| **Version:** | 1.0 |

## Definition

*A token class that implements this behavior will support the burning or decommissioning of token instances of the class. This does not delete a token, but rather places it in a permanent non-use state.  Burning is a one way operation and cannot be reversed. This behavior is Delegable. If the token definition is Delegable, BurnFrom will be available.*

## Example

When a token is used in a certain way, you may want to remove it from circulation or from being used again. Since the ledger doesn't allow for deletions, burning a token essentially 'deletes' the token from being used, but not from history.

## Analogies

| Name | Description |
|---|---|
| **Oil Barrels** | If you mint a new token for each barrel of oil created, you may transfer ownership several times until the barrel is refined. The refining process should burn the barrel of oil to remove it from circulation. |
| **Redeem** | A token that is a coupon or single use ticket, should be burned when it is redeemed. |

## Dependencies

| Artifact Type | Symbol | Description |
|---|---|---|

## Incompatible With

| Artifact Type | Symbol | Id |
|---|---|---|

## Influenced By

| Description | Symbol | Applies To |
|---|---|---|
| **Delegable or not, will determine if the BurnFrom Control will be available in the implementation.** | g | [ ] |
| **If Compliance is present, a CheckBurnAllowed request has to be made and verified before a Burn request or a BurnFrom request.** | c | [ ] |

## Artifact Files

| Content Type | File Name | File Content |
|---|---|---|
| **Control** | burnable.proto | |
| **Uml** | burnable.md | |

## Code Map

| Map Type | Name | Platform | Location |
|---|---|---|---|
| **SourceCode** | Open Zeppelin | EthereumSolidity | https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/token/ERC20/ERC20Burnable.sol |

## Implementation Map

| Map Type | Name | Platform | Location |
|---|---|---|---|

## Resource Map

| Map Type | Name | Location | Description |
|---|---|---|---|

## Specification Behavior

# Burnable

## Taxonomy Symbol: b

*A token class that implements this behavior will support the burning or decommissioning of token instances of the class. This does not delete a token, but rather places it in a permanent non-use state. Burning is a one way operation and cannot be reversed. This behavior is Delegable. If the token definition is Delegable, BurnFrom will be available.*

## Example

When a token is used in a certain way, you may want to remove it from circulation or from being used again. Since the ledger doesn't allow for deletions, burning a token essentially 'deletes' the token from being used, but not from history.

## Analogies

| Name | Description |
|------|-------------|
| **Oil Barrels** | If you mint a new token for each barrel of oil created, you may transfer ownership several times until the barrel is refined. The refining process should burn the barrel of oil to remove it from circulation. |
| **Redeem** | A token that is a coupon or single use ticket, should be burned when it is redeemed. |

| Is External: | False |
|--------------|-------|
| **Constructor:** | |

## Burnable responds to these Invocations

### *Burn*

Id: f063dcaa-49f9-4c49-bf0f-2766301e1033

Description: A request to burn a token instance(s) in the class by the owner of the token instance(s). Optional Quantity field in the request.

### Request Message:

BurnRequest

Description: The request to Burn or Retire tokens.

*Request Parameters*

| Name | Value |
|------|-------|
| **Quantity** | The number of tokens to burn, might not apply to the implementation. |

## Response Message

BurnResponse

Description: The response from the request to burn.

*Response Parameters*

| Name | Value |
|------|-------|
| **Confirmation** | A confirmation receipt or error may be returned to the invoker based on the outcome of the burn request |

## *BurnFrom*

Id: 49b53152-3360-426f-9e0a-24a0b4e7c881

Description: Requires Delegable. A request to burn token instance(s) in the class by a party or account that has allowance to do so.  Requires a From and Quantity fields in the request.

## Request Message:

BurnFromRequest

Description: The request to Burn or Retire tokens.

*Request Parameters*

| Name | Value |
|------|-------|
| **From** | AccountId from which tokens are burnt |
| **Quantity** | The number of tokens to burn, might not apply to the implementation. |

## Response Message

BurnFromResponse

Description: The response from the request to burn.

EThaler – ac49c603701963174e669577afa0112bbd0fe5d3fe29c06fc5a6a3cf794afad223

*Response Parameters*

| Name | Value |
|---|---|
| **Confirmation** | A confirmation receipt or error may be returned to the invoker based on the outcome of the burn from request |

# Properties

## Roles

| Type: | Behavior |
|---|---|
| **Name:** | Roles |
| **Id:** | c32726da-9787-4dd8-8de3-d07d1733d0f6 |
| **Visual:** | <i>r</i> |
| **Tooling:** | r |
| **Version:** | 1.0 |

## Definition

*A token can have behaviors that the class will restrict invocations to a select set of parties or accounts that are members of a role or group. This is a generic behavior that can apply to a token many times to represent many role definitions within the template. This behavior will allow you to define what role(s) to create and what behavior(s) to apply the role to in the TemplateDefinition.*

## Example

## Analogies

| Name | Description |
|---|---|
| **Minters** | A role called 'Minters' for a token can have accounts in the role. The MintTo behavior invocation will be bound to the role check to ensure only account in the 'Minters' role are allowed to mint new instances in the class. |

# Comments

Roles has a constructor control that creates roles and applies them to certain behaviors of the token at creation of the class from the template.

# Dependencies

| Artifact Type | Symbol | Description |
|---|---|---|
| | | |

# Incompatible With

| Artifact Type | Symbol | Id |
|---|---|---|
| | | |

# Influenced By

| Description | Symbol | Applies To |
|---|---|---|
| | | |

# Artifact Files

| Content Type | File Name | File Content |
|---|---|---|
| **Control** | roles.proto | |
| **Uml** | roles.md | |

# Code Map

| Map Type | Name | Platform | Location |
|---|---|---|---|
| **SourceCode** | Open Zeppelin - Roles | EthereumSolidity | https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/access/Roles.sol |

# Implementation Map

| Map Type | Name | Platform | Location |
|---|---|---|---|
| | | | |

# Resource Map

| Map Type | Name | Location | Description |
|---|---|---|---|
| | | | |

Specification Behavior

# Roles

## Taxonomy Symbol: r

*A token can have behaviors that the class will restrict invocations to a select set of parties or accounts that are members of a role or group. This is a generic behavior that can apply to a token many times to represent many role definitions within the template. This behavior will allow you to define what role(s) to create and what behavior(s) to apply the role to in the TemplateDefinition.*

## Example

## Analogies

| Name | Description |
|------|-------------|
| **Minters** | A role called 'Minters' for a token can have accounts in the role. The MintTo behavior invocation will be bound to the role check to ensure only account in the 'Minters' role are allowed to mint new instances in the class. |

## Comments

Roles has a constructor control that creates roles and applies them to certain behaviors of the token at creation of the class from the template.

| Is External: | False |
|--------------|-------|
| **Constructor:** | |

## <u>Roles responds to these Invocations</u>

### *RoleCheck*

Id: 00a665e3-1dda-441e-8262-5750435c153c

Description: Internal invocation when the applied behavior is called to check if the requestor is a member of the role.

### Request Message:

IsInRole

Description: The request

EThaler – ac49c603701963174e669577afa0112bbd0fe5d3fe29c06fc5a6a3cf794afad226

| Name | Value |
|------|-------|
| **AccountId** | AccountId of the requestor. |

## Response Message

True/False

Description: The response

| Name | Value |
|------|-------|
| **IsInRole** | True/False |

# Properties

## *Name: Role*

Value Description: A group or list an account can be a member or be in.

Template Value: Minters

# Invocations

## *GetRoleMembers*

Id:

Description: Request the the list of member accounts in the role.

## Request

Control Message: GetRoleMembersRequest

Description: The request

| Name | Value |
|------|-------|
| | |

## Response

Control Message: GetRoleMembersResponse

Description: The response

EThaler - ac49c603701963174e669577afa0112bbd0fe5d3fe29c06fc5a6a3cf794afad227

| Name | Value |
|------|-------|
| **Members** | Returning the list of accounts in the role. |

## *AddRoleMember*

Id: 600357f8-0499-47f8-87a5-eedf4ad034af

Description: Add a member to the group or role property.

## Request

Control Message: AddRoleMemberRequest

Description: The request

*Parameters*

| Name | Value |
|------|-------|
| **RoleName** | Name of the role you are adding a member to. Optional parameter if there is only one role. |
| **AccountAddress** | Address, name or identifier of the account to be added to the role. |

## Response

Control Message: AddRoleMemberResponse

Description: The response

*Parameters*

| Name | Value |
|------|-------|
| **Added** | True or False. |

## *RemoveRoleMember*

Id: 97e160bb-6c60-4f1d-923b-813b07b89638

Description: Remove a member to the group or role property.

## Request

Control Message: RemoveRoleMemberRequest

Description: The request

| Name | Value |
|------|-------|
| **RoleName** | Name of the role you are adding a member to. Optional parameter if there is only one role. |
| **AccountAddress** | Address, name or identifier of the account to be removed from the role. |

# Response

Control Message: RemoveRoleMemberResponse

Description: The response

*Parameters*

| Name | Value |
|------|-------|
| **Added** | True or False. |

# *IsInRole*

Id: e42b1b16-074a-4d7d-b9f9-f69a2397a21b

Description: Check to see if an account is in the role.

## Request

Control Message: IsInRoleRequest

Description: The request may be internal only and not exposed externally.

*Parameters*

| Name | Value |
|------|-------|
| **RoleName** | Name of the role you are checking membership of. Optional parameter if there is only one role. |
| **AccountAddress** | Address, name or identifier of the account to be checked. |

## Response

Control Message: IsInRoleRequestResponse

Description: The response

| Name | Value |
|------|-------|
| InRole | True or False. |

## GetMinters

Id:

Description: Request the the list of member accounts in the 'Minters' role.

### Request

Control Message: GetMintersRequest

Description: The request

*Parameters*

| Name | Value |
|------|-------|
| | |

### Response

Control Message: GetMintersResponse

Description: The response

*Parameters*

| Name | Value |
|------|-------|
| Members | Returning the list of accounts in the 'Minters' role. |

## AddRoleMember

Id: 600357f8-0499-47f8-87a5-eedf4ad034af

Description: Add a member to the group or role property.

### Request

Control Message: AddRoleMemberRequest

Description: The request

*Parameters*

| Name | Value |
|------|-------|
| RoleName | Value is always set to 'Minters' |

| | |
|---|---|
| **AccountAddress** | Address, name or identifier of the account to be added to the 'Minters' role. |

## Response

Control Message: AddRoleMemberResponse

Description: The response

### *Parameters*

| Name | Value |
|---|---|
| **Added** | True or False. |

## *RemoveRoleMember*

Id: 97e160bb-6c60-4f1d-923b-813b07b89638

Description: Remove a member to the group or role property.

## Request

Control Message: RemoveRoleMemberRequest

Description: The request

### *Parameters*

| Name | Value |
|---|---|
| **RoleName** | Always set to 'Minters' |
| **AccountAddress** | Address, name or identifier of the account to be removed from the role. |

## Response

Control Message: RemoveRoleMemberResponse

Description: The response

### *Parameters*

| Name | Value |
|---|---|
| **Added** | True or False. |

## IsInRole

Id: e42b1b16-074a-4d7d-b9f9-f69a2397a21b

Description: Check to see if an account is in the role.

### Request

Control Message: IsInRoleRequest

Description: The request may be internal only and not exposed externally.

#### Parameters

| Name | Value |
|------|-------|
| RoleName | Always be bound to 'Minters' |
| AccountAddress | Address, name or identifier of the account to be checked. |

### Response

Control Message: IsInRoleRequestResponse

Description: The response

#### Parameters

| Name | Value |
|------|-------|
| InRole | True or False. |

# Properties

## Mintable

| Type: | Behavior |
|-------|----------|
| Name: | Mintable |
| Id: | f9224e90-3cab-45bf-b5dc-0175121e2ead |
| Visual: | <i>m</i> |
| Tooling: | m |
| Version: | 1.0 |

## Definition

*A token class that implements this behavior will support the minting or issuing of new token instances in the class. These new tokens can be minted and*

EThaler – ac49c603701963174e669577afa0112bbd0fe5d3fe29c06fc5a6a3cf794afad232

*belong to the owner or minted to another account. This behavior may be invalidated by a restrictive behavior like Singleton, where only a single instance of the token can exist. Mintable is technically delegable, but it's delegation should be controlled by a behavior like Roles.*

## Example

A consortium of oil producers needs to create tokens for each barrel of oil they are putting on the market to trade. There are separate classes of tokens for each grade of oil. Producers of barrels will need be have the ability to mint new tokens in order to facilitate the trading of them in the supply chain.

## Analogies

| Name | Description |
|------|-------------|
| SKU | A token class can represent a particular item SKU, where the manufacturer of the item has the ability to mint or issue new inventory of the SKU into the supply chain. |

## Dependencies

| Artifact Type | Symbol | Description |
|---------------|--------|-------------|

## Incompatible With

| Artifact Type | Symbol | Id |
|---------------|--------|-----|

## Influenced By

| Description | Symbol | Applies To |
|-------------|--------|-----------|
| **Roles is common to implement to provide authorization checks for invoking the behavior. Highly Recommended that Role restrictions be applied to MintTo invocations.** | r | [ ] |
| **If Compliance is present, a CheckMintAllowed request has to be made and verified before a Mint request or a MintTo request.** | c | [ ] |

## Artifact Files

| Content Type | File Name | File Content |
|--------------|-----------|--------------|
| **Control** | mintable.proto | |

| Uml | mintable.md | |
|---|---|---|

## Code Map

| Map Type | Name | Platform | Location |
|---|---|---|---|
| **SourceCode** | Open Zeppelin | EthereumSolidity | https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/token/ERC20/ERC20Mintable.sol |

## Implementation Map

| Map Type | Name | Platform | Location |
|---|---|---|---|
| **Implementation** | Implementation 1 | ChaincodeGo | |

## Resource Map

| Map Type | Name | Location | Description |
|---|---|---|---|
| **Resource** | Regulation Reference 1 | | |

## Specification Behavior

# Mintable

## Taxonomy Symbol: m

*A token class that implements this behavior will support the minting or issuing of new token instances in the class. These new tokens can be minted and belong to the owner or minted to another account. This behavior may be invalidated by a restrictive behavior like Singleton, where only a single instance of the token can exist. Mintable is technically delegable, but it's delegation should be controlled by a behavior like Roles.*

## Example

A consortium of oil producers needs to create tokens for each barrel of oil they are putting on the market to trade. There are separate classes of tokens for each grade of oil. Producers of barrels will need be have the ability to mint new tokens in order to facilitate the trading of them in the supply chain.

# Analogies

| Name | Description |
|------|-------------|
| SKU | A token class can represent a particular item SKU, where the manufacturer of the item has the ability to mint or issue new inventory of the SKU into the supply chain. |

| | |
|------|------|
| **Is External:** | **False** |
| **Constructor:** | |

# Mintable responds to these Invocations

Binding Is Influenced by Roles's Invocation RoleCheckRoles's Invocation RoleCheck Intercepts this behavior's invocation.'

## *RoleCheck*

Id: 00a665e3-1dda-441e-8262-5750435c153c

Description: Check to see if the account is in the Role called 'Minters'

## Request Message:

IsInRole

Description: Checking the 'Minters' role.

### *Request Parameters*

| Name | Value |
|------|-------|
| **AccountId** | AccountId of the requestor. |

## Response Message

True/False

Description: Respond true if the account is in the 'Minters' role.

### *Response Parameters*

| Name | Value |
|------|-------|
| **IsInRole** | True/False |

## MintTo

Id: 70499b23-a1dd-4c87-90d6-6e45400f28b5

Description: A request to create new token instances in the class by the owner or a party or account in a role that is granted this permission to another party or account.  Requires a To and Quantity fields in the request.

## Request Message:

MintToRequest

Description: The request

### Request Parameters

| Name | Value |
|------|-------|
| ToAccount | Account Id to mint the tokens to. |
| Quantity | Number of new tokens to create. |

## Response Message

MintToResponse

Description: The response

### Response Parameters

| Name | Value |
|------|-------|
| Confirmation | A confirmation receipt or error may be returned to the invoker based on the outcome of the MintTo request. |

## Mint

Id: 3ddf15db-c919-4f72-a57b-d089931bc901

Description: A request to create new token instances in the class by the owner or a party or account in a role that is granted this permission. Minted tokens using this invocation will be owned by the owner or token pool account. Requires a Quantity field in the request.

## Request Message:

MintRequest

Description: The request

EThaler - ac49c603701963174e669577afa0112bbd0fe5d3fe29c06fc5a6a3cf794afad236

| Name | Value |
|------|-------|
| **Quantity** | Number of new tokens to create. |

## Response Message

MintResponse

Description: The response

*Response Parameters*

| Name | Value |
|------|-------|
| **Confirmation** | A confirmation receipt or error may be returned to the invoker based on the outcome of the mint request. |

# Properties

## Supply Control

| Type: | BehaviorGroup |
|-------|---------------|
| **Name:** | Supply Control |
| **Id:** | 91cb89b6-a2ce-44ff-b3a0-f0cb3f117e56 |
| **Visual:** | <i>SC</i> |
| **Tooling:** | SC |
| **Version:** | 1.0 |

## Definition

*A token class that implements this behavior will provide controls to increase and decrease supply of tokens within the class. Additionally, it will include the ability to support a role, like Minters, that will be allowed to invoke the Mintable behavior. The owner can add accounts to the role and any account that is a member of the role will be able to mint tokens in the class.*

## Example

EThaler - ac49c603701963174e669577afa0112bbd0fe5d3fe29c06fc5a6a3cf794afad237

## Analogies

| Name | Description |
|------|-------------|
| **Central Bank** | Implementing monetary policy for this token. |

## Comments

Define a Minters role and apply the role to the mintable behavior.

## Dependencies

| Artifact Type | Symbol | Description |
|---------------|--------|-------------|

## Incompatible With

| Artifact Type | Symbol | Id |
|---------------|--------|----|
| **Behavior** | s | c1189d7a-e142-4504-bf26-44c35b76c9d6 |

## Influenced By

| Description | Symbol | Applies To |
|-------------|--------|------------|
| **Create a Minters Role and apply it to the Mintable behavior to provide authorization checks for invoking the behavior.** | r | [ ] |

## Artifact Files

| Content Type | File Name | File Content |
|--------------|-----------|--------------|
| **Control** | supply-control.proto | |
| **Uml** | supply-control.md | |

## Code Map

| Map Type | Name | Platform | Location |
|----------|------|----------|----------|

## Implementation Map

| Map Type | Name | Platform | Location |
|----------|------|----------|----------|

# Resource Map

| Map Type | Name | Location | Description |
|----------|------|----------|-------------|

*The behaviors belonging to this group are included in the Behaviors section of this specification.*