EEA-REPUTATION

Taxonomy Formula: tF{~d,~t,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 as long as they are in the same class or series.
Token Unit:	Whole	There can be many instances of this token, but they cannot be subdivided.
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 owners 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.

EEA Reputation Tokens are issued, upon vesting, to an organization's contributors establishing an individuals reputation. The token grant should be adjusted when commitments are met or before vesting indicating the split of reputation tokens by percentage to the contributors listed in the grant. The reputation split between contributors is finalized when the grant vests. Both Reward and Penalty tokens are matched 1-1 towards Reputation with the ability to improve or damage an individual's reputation. An individual's reputation cannot be negative so penalties will subtract 1-1 until exhausted or the account balance reaches 0. The reputation score of an organization is the sum of their contributors balances. These tokens are are lifetime tokens and are not

transferable for any member that has earned them. EEA Reputation tokens are minted and burned, but are not redeemable.

Example

For example, if an organization collects 10,000 tokens during its annual membership cycle, they can redeem the EEA Rewards tokens for say $10,000 credit to its membership, or continue to accumulate. In addition, if the organization's lifetime membership EEA Reputation tokens total was 100,000 at the beginning of the membership cycle, it would be 110,000 at the end of the cycle in this example. In addition, 10,000 points would be split across the organization's employees who earned them.

Analogies

Name	Description
Earned Credits	A customer can earn a point/token for each mile travelled and then redeem these points/tokens for upgrades or new tickets, but cannot transfer the points to another party.

Comments

EEA-Reputation is:

- Non-Subdividable
- Non-transferable
- Burnable
- Roles
- Mintable

EEA-Reputation Details

Base: Whole Fungible

Type:	Base
Name:	Whole Fungible
Id:	b1eacdf8-35d8-454a-b1af-92eb0b6f45d4
Visual:	τ_F{~d}
Tooling:	tF{~d}
Version:	1.0

Definition

Whole 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. A whole token cannot be sub-divided so it doesn't support the notion of 'making change'.

Example

An inventory item or SKU, where an item is treated as a whole because it makes no sense to own a fraction of a SKU or loyalty point.

Analogies

Name	Description
Loyalty Points	Most credit card or retail loyalty point programs deal with whole numbers so that redeeming points is easy to understand for their customers.
General Admission Movie Ticket	Purchasing a general admission ticket to a movie only allows for you to have a seat, but the seat that you actually get depends on factors like when you arrive. Your not likely to want to share a seat with another adult.

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	whole-fungible.proto	
Uml	whole-fungible.md	
Other	.DS_Store	

Code Map

Map Type	Name	Platform	Location
SourceCode	Solidity Reward Token	EthereumSolidity	https://github.com/EntEthAlliance/Trusted-Token/blob/develop/contracts/RewardToken.sol
SourceCode	Solidity Reputation Token	EthereumSolidity	https://github.com/EntEthAlliance/Trusted-Token/blob/develop/contracts/ReputationToken.sol

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:	Whole
Symbol:	
Owner:	
Quantity:	0
Decimals:	0
Constructor Name:	Constructor

Behaviors

Base: Non-Subdividable

Type:	**Behavior**
Name:	Non-Subdividable
Id:	d5807a8e-879b-4885-95fa-f09ba2a22172
Visual:	~d
Tooling:	~d
Version:	1.0

Definition

An ability or restriction on the token where it cannot be subdivided from a single whole token into fractions. Sets the base token Decimals property to 0 which will make the token non-sub-dividable and a whole token is the smallest ownable unit of the token.

Example

Non-subdividable is common for items where subdivision does not make sense, like a property title, inventory item or invoice.

Analogies

Name	**Description**
Non-Fractional	It is not possible to own a fraction of this token.
Barrel of Oil	Barrels of Oil don't make sense to subdivide.

Dependencies

Artifact Type	Symbol	Description

Incompatible With

Artifact Type	Symbol	Id
Behavior	d	6e3501dc-5800-4c71-b59e-ad11418a998c

Influenced By

Description	Symbol	Applies To

Artifact Files

Content Type	File Name	File Content
Control	non-subdividable.proto	
Uml	non-subdividable.md	
Other	.DS_Store	

Code Map

Map Type	Name	Platform	Location
SourceCode	Code 1	Daml	

Implementation Map

Map Type	Name	Platform	Location
Implementation	Implementation 1	ChaincodeGo	

Resource Map

Map Type	Name	Location	Description
Resource	Regulation Reference 1		

Specification Behavior

Non-Subdividable

Taxonomy Formula: ~d

An ability or restriction on the token where it cannot be subdivided from a single whole token into fractions. Sets the base token Decimals property to 0 which will make the token non-sub-dividable and a whole token is the smallest ownable unit of the token.

Example

Non-subdividable is common for items where subdivision does not make sense, like a property title, inventory item or invoice.

Analogies

Name	Description
Non-Fractional	It is not possible to own a fraction of this token.
Barrel of Oil	Barrels of Oil don't make sense to subdivide.

Comments

Is External:	True
Constructor:	

Non-Subdividable responds to these Invocations

Properties

Name: Decimals

Value Description: Set to Zero, not allowing any subdivision, usually this is applied to the base token.

Template Value: 0

Invocations

GetDecimals

Id: 2ca7fbb2-ce98-4dda-a6ae-e4ac2527bb33

Description: Should return 0

Request

Control Message: GetDecimalsRequest

Description:

Parameters

Name	Value

Response

Control Message: GetDecimalsResponse

Description: Return 0

Parameters

Name	Value
Decimals	0

GetDecimals

Id: 2ca7fbb2-ce98-4dda-a6ae-e4ac2527bb33

Description: Should return 0

Request

Control Message: GetDecimalsRequest

Description:

Parameters

Name	Value

Response

Control Message: GetDecimalsResponse

Description: Return 0

Parameters

Name	Value
Decimals	0

Properties

Base: Non-transferable

Type:	Behavior
Name:	Non-transferable
Id:	a4fa4ca8-6afd-452b-91f5-7103b6fee5e5
Visual:	~t
Tooling:	~t
Version:	1.0

Definition

Every token instance has an owner. The Non-transferable behavior prevents the owner of a token from changing.

Example

A vote token, for a citizen in a public election would be non-transferable.

Analogies

Name	Description

Diploma	A diploma from an educational institution is not transferable to another party that can claim to have earned the diploma.
Airline Ticket	Due to security restrictions at airports and airlines, tickets can only be used by the person they were issued to.

Dependencies

Artifact Type	Symbol	Description

Incompatible With

Artifact Type	Symbol	Id
Behavior	t	af119e58-6d84-4ca6-9656-75e8d312f038

Influenced By

Description	Symbol	Applies To

Artifact Files

Content Type	File Name	File Content
Control	non-transferable.proto	
Uml	non-transferable.md	
Other	.DS_Store	

Code Map

Map Type	Name	Platform	Location
SourceCode	Code 1	Daml	

Implementation Map

Map Type	Name	Platform	Location
Implementation	Implementation 1	ChaincodeGo	

Resource Map

Map Type	Name	Location	Description
Resource	Regulation Reference 1		

Specification Behavior

Non-transferable

Taxonomy Formula: ~t

Every token instance has an owner. The Non-transferable behavior prevents the owner of a token from changing.

Example

A vote token, for a citizen in a public election would be non-transferable.

Analogies

Name	Description
Diploma	A diploma from an educational institution is not transferable to another party that can claim to have earned the diploma.
Airline Ticket	Due to security restrictions at airports and airlines, tickets can only be used by the person they were issued to.

Comments

Is External:	True
Constructor:	

Non-transferable responds to these Invocations

Properties

Base: Burnable

Type:	Behavior
Name:	Burnable
Id:	803297a1-c0f9-4898-9d44-29c9d41cca97
Visual:	b
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	
Other	.DS_Store	

Code Map

Map Type	Name	Platform	Location
SourceCode	OpenZeppelin	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
Resource	Regulation Reference 1		

Specification Behavior

Burnable

Taxonomy Formula: 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.

Comments

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.

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

Base: Roles

Type:	Behavior
Name:	Roles
Id:	c32726da-9787-4dd8-8de3-d07d1733d0f6
Visual:	r
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	File Name	File Content

Type		
Control	roles.proto	
Uml	roles.md	
Other	.DS_Store	

Code Map

Map Type	Name	Platform	Location
SourceCode	Code 1	Daml	

Implementation Map

Map Type	Name	Platform	Location
Implementation	Implementation 1	ChaincodeGo	

Resource Map

Map Type	Name	Location	Description
Resource	Regulation Reference 1		

Specification Behavior

Roles

Taxonomy Formula: 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:	

Roles responds to these Invocations

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

Request Parameters

Name	Value
AccountId	AccountId of the requestor.

Response Message

True/False

Description: The response

Response Parameters

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

Parameters

Name	Value

Response

Control Message: GetRoleMembersResponse

Description: The response

Parameters

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

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 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

Parameters

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

Base: Mintable

Type:	Behavior
Name:	Mintable
Id:	f9224e90-3cab-45bf-b5dc-0175121e2ead

Visual:	m
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 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	
Other	.DS_Store	

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 Formula: 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.

Comments

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

Request Parameters

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

Base: Supply Control

Type:	BehaviorGroup
Name:	Supply Control
Id:	91cb89b6-a2ce-44ff-b3a0-f0cb3f117e56
Visual:	SC
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

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	
Other	.DS_Store	

Code Map

Map Type	Name	Platform	Location

Implementation Map

Map Type	Name	Platform	Location

Resource Map

Map Type	Name	Location	Description

Behavior Group Details

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