

# IEEE Standard for Blockchainbased Digital Asset Identification

---

## IEEE Standard for Blockchainbased Digital Asset Identification

Overview

Background

Scope

Purpose

Normative references

Definitions, abbreviations and acronyms

Definitions

Abbreviations and acronyms

Specification

Methods

Data structures

Others

Bibliography

## Overview

---

### Background

The asset identification specification is the key to establishing a digital asset management system, especially when it comes to multi-asset management and cross-chain asset operations. Without a universal digital asset identification specification, asset management based on different protocols will become more and more complicated.

### Scope

- Define the data structure related to digital asset identification;
- Define data format specifications related to digital asset identification;
- Propose asset management operation specifications related to digital asset identification.

### Purpose

This standard defines the data structure related to asset identification to improve the digital asset management efficiency, provide guidance for the design of digital asset management solutions, and provide a reference for building a digital asset service platform.

### Normative references

---

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

### Definitions, abbreviations and acronyms

---

## Definitions

- **Blockchain:** Distributed ledger with confirmed blocks organized in an append-only, sequential chain using cryptographic links.

NOTE-See [\[B1\]](#)

- **Digital asset:** Asset that exist only in digital form or which is the digital representation of another asset.

NOTE-See [\[B1\]](#)

- **Token:** Digital asset that represents a collection of entitlements.

NOTE-See [\[B1\]](#)

- **Fungible Token:** Fungible Tokens have a property that makes each Token be exactly the same (in type and value) of another Token.

NOTE-See [\[B2\]](#)

- **Non-Fungible Token:** A Non-Fungible Token (NFT) is used to identify something or someone in a unique way. NOTE-See [\[B3\]](#)

## Abbreviations and acronyms

- **FT:** Fungible Token.
- **NFT:** Non-Fungible Token.

## Specification

---

### Methods

Method Name	Request Type	Response Type	Description
Create	<a href="#">CreateInput</a>	<a href="#">Empty</a>	Create a new token.
Issue	<a href="#">IssueInput</a>	<a href="#">Empty</a>	Issuing some amount of tokens to an address is the action of increasing that addresses balance for the given token. The total amount of issued tokens must not exceed the total supply of the token and only the issuer (creator) of the token can issue tokens. Issuing tokens effectively increases the circulating supply.
Transfer	<a href="#">TransferInput</a>	<a href="#">Empty</a>	Transferring tokens simply is the action of transferring a given amount of tokens from one address to another. The origin or source address is the signer of the transaction. The balance of the sender must be higher than the amount that is transferred.
TransferFrom	<a href="#">TransferFromInput</a>	<a href="#">Empty</a>	The TransferFrom action will transfer a specified amount of tokens from one address to another. For this operation to succeed the from address needs to have approved (see allowances) enough tokens to Sender of this transaction. If successful the amount will be removed from the allowance.

Method Name	Request Type	Response Type	Description
BatchTransferFrom	<a href="#">BatchTransferFromInput</a>	<a href="#">Empty</a>	The BatchTransferFrom action will transfer a batch of specified amount of tokens from one address to another. For this operation to succeed the from address needs to have approved (see allowances) enough tokens to Sender of this transaction. If successful the amount will be removed from the allowance.
Approve	<a href="#">ApproveInput</a>	<a href="#">Empty</a>	The approve action increases the allowance from the Sender to the Spender address, enabling the Spender to call TransferFrom.
UnApprove	<a href="#">UnApproveInput</a>	<a href="#">Empty</a>	This is the reverse operation for Approve, it will decrease the allowance.
Lock	<a href="#">LockInput</a>	<a href="#">Empty</a>	This method can be used to lock tokens.
Unlock	<a href="#">UnlockInput</a>	<a href="#">Empty</a>	This is the reverse operation of locking, it un-locks some previously locked tokens.
Burn	<a href="#">BurnInput</a>	<a href="#">Empty</a>	This action will burn the specified amount of tokens, removing them from the token's Supply.
ChangeTokenIssuer	<a href="#">ChangeTokenIssuerInput</a>	<a href="#">Empty</a>	Change the issuer of the specified token. Only the original issuer can change it.
SetPrimaryTokenSymbol	<a href="#">SetPrimaryTokenSymbolInput</a>	<a href="#">Empty</a>	Set the primary token of side chain.
CrossChainTransfer	<a href="#">CrossChainTransferInput</a>	<a href="#">Empty</a>	This interface is used for cross-chain transfer.
CrossChainReceiveToken	<a href="#">CrossChainReceiveTokenInput</a>	<a href="#">Empty</a>	This method is used to receive cross-chain transfers.
CrossChainCreateToken	<a href="#">CrossChainCreateTokenInput</a>	<a href="#">Empty</a>	The side chain creates tokens.
InitializeFromParentChain	<a href="#">InitializeFromParentChainInput</a>	<a href="#">Empty</a>	When the side chain is started, the side chain is initialized with the parent chain information.
ClaimTransactionFees	<a href="#">TotalTransactionFeesMap</a>	<a href="#">Empty</a>	Handle the transaction fees charged by ChargeTransactionFees.
ChargeTransactionFees	<a href="#">ChargeTransactionFeesInput</a>	<a href="#">ChargeTransactionFeesOutput</a>	Used to collect transaction fees.
CheckThreshold	<a href="#">CheckThresholdInput</a>	<a href="#">Empty</a>	Check the token threshold.
InitialCoefficients	<a href="#">Empty</a>	<a href="#">Empty</a>	Initialize coefficients of every type of tokens supporting charging fee.
DonateResourceToken	<a href="#">TotalResourceTokensMaps</a>	<a href="#">Empty</a>	Processing resource token received.
ChargeResourceToken	<a href="#">ChargeResourceTokenInput</a>	<a href="#">Empty</a>	A transaction resource fee is charged to implement the ACS8 standards.
CheckResourceToken	<a href="#">Empty</a>	<a href="#">Empty</a>	Verify that the resource token are sufficient.
SetSymbolsToPayTxSizeFee	<a href="#">SymbolListToPayTxSizeFee</a>	<a href="#">Empty</a>	Set the list of tokens to pay transaction fees.
UpdateCoefficientsForSender	<a href="#">UpdateCoefficientsInput</a>	<a href="#">Empty</a>	Update the coefficient of the transaction fee calculation formula.
UpdateCoefficientsForContract	<a href="#">UpdateCoefficientsInput</a>	<a href="#">Empty</a>	Update the coefficient of the transaction fee calculation formula.
InitializeAuthorizedController	<a href="#">Empty</a>	<a href="#">Empty</a>	This method is used to initialize the governance organization for some functions, including: the coefficient of the user transaction fee calculation formula, the coefficient of the contract developer resource fee calculation formula, and the side chain rental fee.
GetTokenInfo	<a href="#">GetTokenInfoInput</a>	<a href="#">TokenInfo</a>	Query token information.
GetNativeTokenInfo	<a href="#">Empty</a>	<a href="#">TokenInfo</a>	Query native token information.
GetResourceTokenInfo	<a href="#">Empty</a>	<a href="#">TokenInfoList</a>	Query resource token information.
GetBalance	<a href="#">GetBalanceInput</a>	<a href="#">GetBalanceOutput</a>	Query the balance at the specified address.

Method Name	Request Type	Response Type	Description
GetBalanceBatch	<a href="#">GetBalanceBatchInput</a>	<a href="#">GetBalanceBatchOutput</a>	Batch query the balance at the specified address.
GetAllowance	<a href="#">GetAllowanceInput</a>	<a href="#">GetAllowanceOutput</a>	Query the account's allowance for other addresses
IsInWhiteList	<a href="#">IsInWhiteListInput</a>	<a href="#">BoolValue</a>	Check whether the token is in the whitelist of an address, which can be called TransferFrom to transfer the token under the condition of not being credited.
GetLockedAmount	<a href="#">GetLockedAmountInput</a>	<a href="#">GetLockedAmountOutput</a>	Query the information for a lock.
GetCrossChainTransferTokenContractAddress	<a href="#">GetCrossChainTransferTokenContractAddressInput</a>	<a href="#">Address</a>	Query the address of receiving token in cross-chain transfer.
GetPrimaryTokenSymbol	<a href="#">Empty</a>	<a href="#">StringValue</a>	Query the name of the primary Token.
GetCalculateFeeCoefficientsForContract	<a href="#">Int32Value</a>	<a href="#">CalculateFeeCoefficients</a>	Query the coefficient of the transaction fee calculation formula.
GetCalculateFeeCoefficientsForSender	<a href="#">Empty</a>	<a href="#">CalculateFeeCoefficients</a>	Query the coefficient of the transaction fee calculation formula.
GetSymbolsToPayTxSizeFee	<a href="#">Empty</a>	<a href="#">SymbolListToPayTxSizeFee</a>	Query tokens that can pay transaction fees.
GetLatestTotalTransactionFeesMapHash	<a href="#">Empty</a>	<a href="#">Hash</a>	Query the hash of the last input of ClaimTransactionFees.
GetLatestTotalResourceTokensMapsHash	<a href="#">Empty</a>	<a href="#">Hash</a>	Query the hash of the last input of DonateResourceToken.
IsTokenAvailableForMethodFee	<a href="#">StringValue</a>	<a href="#">BoolValue</a>	
CrossChainTransferBatch	<a href="#">CrossChainTransferBatchInput</a>	<a href="#">Empty</a>	This interface is used for batch cross-chain transfer.

## Data structures

- **Address** The function of address return. Output parameters: address.

Field	Type	Label	Description
address	<a href="#">string</a>		Address.

- **AllCalculateFeeCoefficients** The function of Calculating all fees. Output parameters: value.

Field	Type	Label	Description
value	<a href="#">CalculateFeeCoefficients</a>	repeated	The coefficients of fee Calculation.

- **ApproveForAllInput** The function of increasing the allowance of all one's tokens from the Sender to the Spender address, Input parameters: spender, symbol.

Field	Type	Label	Description
spender	<a href="#">Address</a>		The address of an account/contract that is approved to make the operate.
symbol	<a href="#">string</a>		The symbol of token to approve.

- **ApproveInput** The function of increasing the allowance from the Sender to the Spender address, Input parameters: spender, symbol, amount.

Field	Type	Label	Description
spender	<a href="#">Address</a>		The address that allowance will be increased.
symbol	<a href="#">string</a>		The symbol of token to approve.
amount	<a href="#">int64</a>		The amount of token to approve.

- **Approved** The event of increasing the allowance from the Sender to the Spender address. Input parameters: owner, spender, symbol, amount.

Field	Type	Label	Description
owner	<a href="#">Address</a>		The address of the token owner.
spender	<a href="#">Address</a>		The address that allowance be increased.
symbol	<a href="#">string</a>		The symbol of approved token.
amount	<a href="#">int64</a>		The amount of approved token.

- **BatchTransferFromInput** The function of batch transferring form. Input parameters: from, to, symbol, amount, memo.

Field	Type	Label	Description
from	<a href="#">Address</a>		The source address of the token.
to	<a href="#">Address</a>		The destination address of the token.
symbol	<a href="#">string</a>	repeated	The symbol of the token to transfer.
amount	<a href="#">int64</a>	repeated	The amount to transfer.
memo	<a href="#">string</a>		The memo.

- **BoolValue** The function of BoolValue return. Output parameters: bool\_value.

Field	Type	Label	Description
-------	------	-------	-------------

Field	Type	Label	Description
bool_value	<a href="#">bool</a>		Bool value.

- **BurnInput** The function of burning the specified amount of tokens, removing them from the token's Supply. Input parameters: symbol, amount, memo.

Field	Type	Label	Description
symbol	<a href="#">string</a>		The symbol of token to burn.
amount	<a href="#">int64</a>		The amount of token to burn.

- **Burned** The event of burning the specified amount of tokens, removing them from the token's Supply. Input parameters: burner, symbol, amount.

Field	Type	Label	Description
burner	<a href="#">Address</a>		The address who wants to burn token.
symbol	<a href="#">string</a>		The symbol of burned token.
amount	<a href="#">int64</a>		The amount of burned token.

- **CalculateFeeAlgorithmUpdated** The event of updating calculate fee algorithm. Input parameters: all\_type\_fee\_coefficients.

Field	Type	Label	Description
all_type_fee_coefficients	<a href="#">AllCalculateFeeCoefficients</a>		All calculate fee coefficients after modification.

- **CalculateFeeCoefficients** The function of Calculating fees. Output parameters: fee\_token\_type, piece\_coefficients\_list.

Field	Type	Label	Description
fee_token_type	<a href="#">int32</a>		The resource fee type, like READ, WRITE, etc.

Field	Type	Label	Description
piece_coefficients_list	<a href="#">CalculateFeePieceCoefficients</a>	repeated	Coefficients of one single piece.

- **CalculateFeePieceCoefficients** The function of Calculating piece fees. Output parameters: value.

Field	Type	Label	Description
value	<a href="#">int32</a>	repeated	Coefficients of one single piece. The first char is its type: liner / power. The second char is its piece upper bound.

- **ChainPrimaryTokenSymbolSet** The event of setting primary token. Input parameters: token\_symbol.

Field	Type	Label	Description
token_symbol	<a href="#">string</a>		The symbol of token.

- **ChangeTokenIssuerInput** The function of changing the issuer of the specified token. Input parameters: symbol, new\_token\_issuer.

Field	Type	Label	Description
symbol	<a href="#">string</a>		The token symbol.
new_token_issuer	<a href="#">Address</a>		The new token issuer for change.

- **ChargeResourceTokenInput** The function of charging a transaction resource fee(Implemented from the ACS8 standards). Input parameters: cost\_dic, caller.

Field	Type	Label	Description
cost_dic	<a href="#">ChargeResourceTokenInput.CostDicEntry</a>	repeated	Collection of charge resource token, Symbol->Amount.
caller	<a href="#">Address</a>		The sender of the transaction.

- **ChargeResourceTokenInput.CostDicEntry**

Field	Type	Label	Description
key	<a href="#">string</a>		
value	<a href="#">int64</a>		

- **ChargeTransactionFeesInput** The function of collecting transaction fees. Input parameters: method\_name, contract\_address, contract\_address, symbols\_to\_pay\_tx\_size\_fee.

Field	Type	Label	Description
method_name	<a href="#">string</a>		The method name of transaction.
contract_address	<a href="#">Address</a>		The contract address of transaction.
transaction_size_fee	<a href="#">int64</a>		The amount of transaction size fee.
symbols_to_pay_tx_size_fee	<a href="#">SymbolToPayTxSizeFee</a>	repeated	Transaction fee token information.

- **ChargeTransactionFeesOutput** The output of collecting transaction fees. Output parameters: success, charging\_information.

Field	Type	Label	Description
success	<a href="#">bool</a>		Whether the charge was successful.
charging_information	<a href="#">string</a>		The charging information.

- **CheckThresholdInput** The function of checking the token threshold. Input parameters: sender, symbol\_to\_threshold, is\_check\_allowance.

Field	Type	Label	Description
-------	------	-------	-------------



Field	Type	Label	Description
sender	<a href="#">Address</a>		The sender of the transaction.
symbol_to_threshold	<a href="#">CheckThresholdInput.SymbolToThresholdEntry</a>	repeated	The threshold to set, Symbol->Threshold.
is_check_allowance	<a href="#">bool</a>		Whether to check the allowance.

- **CheckThresholdInput.SymbolToThresholdEntry**

Field	Type	Label	Description
key	<a href="#">string</a>		
value	<a href="#">int64</a>		

- **ContractTotalResourceTokens** The function of resource tokens to charge. Input parameters: contract\_address, tokens\_map.

Field	Type	Label	Description
contract_address	<a href="#">Address</a>		The contract address.
tokens_map	<a href="#">TotalResourceTokensMap</a>		Resource tokens to charge.

- **CreateInput** The function of creating a token. Input parameters: symbol, token\_name, supply, total\_supply, decimals, issuer, is\_burnable, issue\_chain\_id, issued, external\_information.

Field	Type	Label	Description
symbol	<a href="#">string</a>		The symbol of the token.
token_name	<a href="#">string</a>		The full name of the token.
total_supply	<a href="#">int64</a>		The total supply of the token.

Field	Type	Label	Description
decimals	<a href="#">int32</a>		The precision of the token.
issuer	<a href="#">Address</a>		The address that created the token.
is_burnable	<a href="#">bool</a>		A flag indicating if this token is burnable.
lock_white_list	<a href="#">Address</a>	repeated	A whitelist address list used to lock tokens.
issue_chain_id	<a href="#">int32</a>		The chain id of the token.
external_information	<a href="#">CreateInput.ExternalInformationEntry</a>	repeated	The external information.

- **CreateInput.ExternalInformationEntry**

Field	Type	Label	Description
key	<a href="#">string</a>		
value	<a href="#">string</a>		

- **CrossChainBatchReceived** The event of batch receiving the token(cross-chain). Input parameters: from, to, symbol, amount, from\_chain\_id, issue\_chain\_id, parent\_chain\_height.

Field	Type	Label	Description
from	<a href="#">Address</a>		The source address of the transferred token.
to	<a href="#">Address</a>		The destination address of the transferred token.
symbol	<a href="#">string</a>	repeated	The symbol of the received token.
amount	<a href="#">int64</a>	repeated	The amount of the received token.

Field	Type	Label	Description
from_chain_id	<a href="#">int32</a>		The destination chain id.
issue_chain_id	<a href="#">int32</a>		The chain id of the token.
parent_chain_height	<a href="#">int64</a>		The parent chain height of the transfer transaction.

- **CrossChainCreateTokenInput** The function of creating tokens on side chain. Input parameters: from\_chain\_id, parent\_chain\_height, transaction\_bytes, merkle\_path.

Field	Type	Label	Description
from_chain_id	<a href="#">int32</a>		The chain id of the chain on which the token was created.
parent_chain_height	<a href="#">int64</a>		The height of the transaction that created the token.
transaction_bytes	<a href="#">bytes</a>		The transaction that created the token.
merkle_path	<a href="#">MerklePath</a>		The merkle path created from the transaction that created the transaction.

- **CrossChainReceiveTokenInput** The function of receiving cross-chain transfers. Input parameters: from\_chain\_id, parent\_chain\_height, transfer\_transaction\_bytes, merkle\_path

Field	Type	Label	Description
from_chain_id	<a href="#">int32</a>		The source chain id.
parent_chain_height	<a href="#">int64</a>		The height of the transfer transaction.
transfer_transaction_bytes	<a href="#">bytes</a>		The raw bytes of the transfer transaction.
merkle_path	<a href="#">MerklePath</a>		The merkle path created from the transfer transaction.

- **CrossChainReceived** The event of receiving the token(cross-chain). Input parameters: from, to, symbol, amount, memo, from\_chain\_id, issue\_chain\_id, parent\_chain\_height.

Field	Type	Label	Description
-------	------	-------	-------------

Field	Type	Label	Description
from	<a href="#">Address</a>		The source address of the transferred token.
to	<a href="#">Address</a>		The destination address of the transferred token.
symbol	<a href="#">string</a>		The symbol of the received token.
amount	<a href="#">int64</a>		The amount of the received token.
memo	<a href="#">string</a>		The memo.
from_chain_id	<a href="#">int32</a>		The destination chain id.
issue_chain_id	<a href="#">int32</a>		The chain id of the token.
parent_chain_height	<a href="#">int64</a>		The parent chain height of the transfer transaction.

- **CrossChainTransferBatchInput** The function of batch transferring(cross-chain). Input parameters: from, to, symbol, amount, to\_chain\_id, issue\_chain\_id, memo.

Field	Type	Label	Description
from	<a href="#">Address</a>		The signer of the transaction.
to	<a href="#">Address</a>		The receiver of transfer.
symbol	<a href="#">string</a>	repeated	The symbol of token.
amount	<a href="#">int64</a>	repeated	The amount of token to transfer.
to_chain_id	<a href="#">int32</a>		The destination chain id.
issue_chain_id	<a href="#">int32</a>		The chain id of the token.
memo	<a href="#">string</a>		The memo.

- **CrossChainTransferInput** The function of cross-chain transferring. Input parameters: to, symbol, amount, memo, to\_chain\_id, issue\_chain\_id.

Field	Type	Label	Description
to	<a href="#">Address</a>		The receiver of transfer.
symbol	<a href="#">string</a>		The symbol of token.
amount	<a href="#">int64</a>		The amount of token to transfer.
memo	<a href="#">string</a>		The memo.

Field	Type	Label	Description
to_chain_id	<a href="#">int32</a>		The destination chain id.
issue_chain_id	<a href="#">int32</a>		The chain id of the token.

- **CrossChainTransferred** The event of transferring tokens(cross-chain). Input parameters: from, to, symbol, amount, to\_chain\_id, issue\_chain\_id, memo.

Field	Type	Label	Description
from	<a href="#">Address</a>		The source address of the transferred token.
to	<a href="#">Address</a>		The destination address of the transferred token.
symbol	<a href="#">string</a>		The symbol of the transferred token.
amount	<a href="#">int64</a>		The amount of the transferred token.
to_chain_id	<a href="#">int32</a>		The destination chain id.
issue_chain_id	<a href="#">int32</a>		The chain id of the token.
memo	<a href="#">string</a>		The memo.

- **CrossChainTransferredBatch** The event of batch transferring the token(cross-chain). Input parameters: from, to, symbol, amount, to\_issue\_chain\_id, issue\_chain\_id.

Field	Type	Label	Description
from	<a href="#">Address</a>		The source address of the transferred token.
to	<a href="#">Address</a>		The destination address of the transferred token.
symbol	<a href="#">string</a>	repeated	The symbol of the transferred token.
amount	<a href="#">int64</a>	repeated	The amount of the transferred token.
to_chain_id	<a href="#">int32</a>		The destination chain id.
issue_chain_id	<a href="#">int32</a>		The chain id of the token.

- **Empty** The function of empty return. Output parameters: empty.

Field	Type	Label	Description
-------	------	-------	-------------

Field	Type	Label	Description
empty	<a href="#">string</a>		Empty return.

- **ExtraTokenListModified** The event of token list modified. Input parameters:  
symbol\_list\_to\_pay\_tx\_size\_fee.

Field	Type	Label	Description
symbol_list_to_pay_tx_size_fee	<a href="#">SymbolListToPayTxSizeFee</a>		Transaction fee token information.

- **GetAllowanceInput** The function of querying the account's allowance for other addresses.  
Input parameters: symbol, owner, spender.

Field	Type	Label	Description
symbol	<a href="#">string</a>		The symbol of token.
owner	<a href="#">Address</a>		The address of the token owner.
spender	<a href="#">Address</a>		The address of the spender.

- **GetAllowanceOutput** The output of querying the account's allowance for other addresses.  
Out parameters: symbol, owner, spender, allowance.

Field	Type	Label	Description
symbol	<a href="#">string</a>		The symbol of token.
owner	<a href="#">Address</a>		The address of the token owner.
spender	<a href="#">Address</a>		The address of the spender.
allowance	<a href="#">int64</a>		The amount of allowance.

- **GetBalanceBatchInput** The function of batch querying the balance at the specified address.  
Input parameters: symbol, owner.

Field	Type	Label	Description
symbol	<a href="#">string</a>	repeated	The symbol of token.

Field	Type	Label	Description
owner	<a href="#">Address</a>		The target address of the query.

- **GetBalanceBatchOutput** The output of batch querying the balance at the specified address. Output parameters: symbol, owner, balance.

Field	Type	Label	Description
symbol	<a href="#">string</a>	repeated	The symbol of token.
owner	<a href="#">Address</a>		The target address of the query.
balance	<a href="#">int64</a>	repeated	The balance of the owner.

- **GetBalanceInput** The function of querying the balance at the specified address. Input parameters: symbol, owner.

Field	Type	Label	Description
symbol	<a href="#">string</a>		The symbol of token.
owner	<a href="#">Address</a>		The target address of the query.

- **GetBalanceOutput** The output of querying the balance at the specified address. Output parameters: symbol, owner, balance.

Field	Type	Label	Description
symbol	<a href="#">string</a>		The symbol of token.
owner	<a href="#">Address</a>		The target address of the query.
balance	<a href="#">int64</a>		The balance of the owner.

- **GetCrossChainTransferTokenContractAddressInput** The function of querying the address of receiving token in cross-chain transfer. Input parameters: chain\_id.

Field	Type	Label	Description
chainId	<a href="#">int32</a>		The chain id.

- **GetLockedAmountInput** The function of querying the information for a lock. Input parameters: address, symbol, lock\_id.

Field	Type	Label	Description
address	<a href="#">Address</a>		The address of the lock.
symbol	<a href="#">string</a>		The token symbol.
lock_id	<a href="#">Hash</a>		The id of the lock.

- **GetLockedAmountOutput** The output of querying the information for a lock. Input parameters: address, symbol, lock\_id, amount.

Field	Type	Label	Description
address	<a href="#">Address</a>		The address of the lock.
symbol	<a href="#">string</a>		The token symbol.
lock_id	<a href="#">Hash</a>		The id of the lock.
amount	<a href="#">int64</a>		The locked amount.

- **GetTokenInfoInput** The function of querying token information. Input parameters: symbol.

Field	Type	Label	Description
symbol	<a href="#">string</a>		The symbol of token.

- **Hash** The function of hash return. Output parameters: hash.

Field	Type	Label	Description
hash	<a href="#">string</a>		Hash value.

- **InitializeFromParentChainInput** The function of initializing side chain with the parent chain's information, when the side chain is start. Input parameters: resource\_amount, registered\_other\_token\_contract\_addresses, creator.



Field	Type	Label	Description
resource_amount	<a href="#">InitializeFromParentChainInput.ResourceAmountEntry</a>	repeated	The amount of resource.
registered_other_token_contract_addresses	<a href="#">InitializeFromParentChainInput.RegisteredOtherTokenContractAddressesEntry</a>	repeated	The token contract addresses.
creator	<a href="#">Address</a>		The creator the side chain.

- **InitializeFromParentChainInput.RegisteredOtherTokenContractAddressesEntry**

Field	Type	Label	Description
key	<a href="#">int32</a>		
value	<a href="#">Address</a>		

- **InitializeFromParentChainInput.ResourceAmountEntry**

Field	Type	Label	Description
key	<a href="#">string</a>		
value	<a href="#">int32</a>		

- **Int32Value** The function of Int32Value return. Output parameters: int\_value.

Field	Type	Label	Description
int_value	<a href="#">int32</a>		Int32 value.

- **IsInWhiteListInput** The functio of checking whether the token is in the whitelist of an address, which can be called TransferFrom to transfer the token under the condition of not being credited. Input parameters: symbol, address.

Field	Type	Label	Description
symbol	<a href="#">string</a>		The symbol of token.
address	<a href="#">Address</a>		The address to check.

- **IssueInput** The function of issuing some amount of tokens to an address. Input parameters: symbol, amount, to, memo.

Field	Type	Label	Description
symbol	<a href="#">string</a>		The token symbol to issue.
amount	<a href="#">int64</a>		The token amount to issue.
memo	<a href="#">string</a>		The memo.
to	<a href="#">Address</a>		The target address to issue.

- **Issued** The event of issuing token. Input parameters: symbol, amount, memo, to.

Field	Type	Label	Description
symbol	<a href="#">string</a>		The symbol of issued token.
amount	<a href="#">int64</a>		The amount of issued token.
memo	<a href="#">string</a>		The memo.
to	<a href="#">Address</a>		The issued target address.

- **LockInput** The function of locking tokens. Input parameters: lock\_address, lock\_id, symbol, amount, memo.

Field	Type	Label	Description
address	<a href="#">Address</a>		The one want to lock his token.
lock_id	<a href="#">Hash</a>		Id of the lock.
symbol	<a href="#">string</a>		The symbol of the token to lock.
memo	<a href="#">string</a>		a memo.
amount	<a href="#">int64</a>		The amount of tokens to lock.

- **MerklePath** The function of merklePath return. Output parameters: merkle\_path.

Field	Type	Label	Description
-------	------	-------	-------------

Field	Type	Label	Description
merkle_path	<a href="#">string</a>		Merkle path.

- **RentalAccountBalanceInsufficient** The event of charging rental fee(which is insufficient).  
Input parameters: symbol, amount.

Field	Type	Label	Description
symbol	<a href="#">string</a>		The symbol of insufficient rental account balance.
amount	<a href="#">int64</a>		The balance of the account.

- **RentalCharged** The event of charging rental fee. Input parameters: symbol, amount.

Field	Type	Label	Description
symbol	<a href="#">string</a>		The symbol of rental fee charged.
amount	<a href="#">int64</a>		The amount of rental fee charged.

- **SetPrimaryTokenSymbolInput** The function of setting the primary token of side chain.  
Input parameters: symbol.

Field	Type	Label	Description
symbol	<a href="#">string</a>		The symbol of the token.

- **StringValue** The function of StringValue return. Output parameters: string\_value.

Field	Type	Label	Description
string_value	<a href="#">string</a>		String value.

- **SymbolListToPayTxSizeFee** The function of transaction fee weight calculation(list). Input parameters: symbols\_to\_pay\_tx\_size\_fee.

Field	Type	Label	Description
symbols_to_pay_tx_size_fee	<a href="#">SymbolToPayTxSizeFee</a>	repeated	Transaction fee token information.

- **SymbolToPayTxSizeFee** The function of transaction fee weight calculation. Input parameters: token\_symbol, base\_token\_weight, added\_token\_weight.

Field	Type	Label	Description
token_symbol	<a href="#">string</a>		The symbol of token.
base_token_weight	<a href="#">int32</a>		The charge weight of primary token.
added_token_weight	<a href="#">int32</a>		The new added token charge weight. For example, the charge weight of primary Token is set to 1. The newly added token charge weight is set to 10. If the transaction requires 1 unit of primary token, the user can also pay for 10 newly added tokens.

- **TokenCreated** The event of creating token. Input parameters: symbol, token\_name, total\_supply, decimals, issuer, is\_burnable, issue\_chain\_id, external information..

Field	Type	Label	Description
symbol	<a href="#">string</a>		The symbol of the token.
token_name	<a href="#">string</a>		The full name of the token.
total_supply	<a href="#">int64</a>		The total supply of the token.
decimals	<a href="#">int32</a>		The precision of the token.
issuer	<a href="#">Address</a>		The address that created the token.
is_burnable	<a href="#">bool</a>		A flag indicating if this token is burnable.
issue_chain_id	<a href="#">int32</a>		The chain id of the token.

Field	Type	Label	Description
external_information	<a href="#">TokenCreated.ExternalInformationEntry</a>	repeated	The external information.

- **TokenCreated.ExternalInformationEntry**

Field	Type	Label	Description
key	<a href="#">string</a>		
value	<a href="#">string</a>		

- **TokenInfo** The information of token. Output parameters: symbol, token\_name, supply, total\_supply, decimals, issuer, is\_burnable, issue\_chain\_id, issued, external\_information.

Field	Type	Label	Description
symbol	<a href="#">string</a>		The symbol of the token.
token_name	<a href="#">string</a>		The full name of the token.
supply	<a href="#">int64</a>		The current supply of the token.
total_supply	<a href="#">int64</a>		The total supply of the token.
decimals	<a href="#">int32</a>		The precision of the token.
issuer	<a href="#">Address</a>		The address that created the token.
is_burnable	<a href="#">bool</a>		A flag indicating if this token is burnable.
issue_chain_id	<a href="#">int32</a>		The chain id of the token.

Field	Type	Label	Description
issued	<a href="#">int64</a>		The amount of issued tokens.
external_information	<a href="#">TokenInfo.ExternalInformationEntry</a>	repeated	The external information.

- **TokenInfo.ExternalInformationEntry**

Field	Type	Label	Description
key	<a href="#">string</a>		
value	<a href="#">string</a>		

- **TokenInfoList** The output of tokeninfo(list). Output parameters: value

Field	Type	Label	Description
value	<a href="#">TokenInfo</a>	repeated	List of token information.

- **TotalResourceTokensMap** The function of processing resource token received. Output parameters: value.

Field	Type	Label	Description
value	<a href="#">TotalResourceTokensMap.ValueEntry</a>	repeated	Resource token dictionary, Symbol->Amount.

- **TotalResourceTokensMap.ValueEntry**

Field	Type	Label	Description
key	<a href="#">string</a>		
value	<a href="#">int64</a>		

- **TotalResourceTokensMaps** The function of processing resource token received. Output parameters: value, block\_hash, block\_height.

Field	Type	Label	Description
value	<a href="#">ContractTotalResourceTokens</a>	repeated	Resource tokens to charge.
block_hash	<a href="#">Hash</a>		The hash of the block processing the transaction.
block_height	<a href="#">int64</a>		The height of the block processing the transaction.

- **TotalSupply** The function of querying the total supply of the token. Input parameters: owner, symbol, amount, memo.

Field	Type	Label	Description
owner	<a href="#">Address</a>		The owner who issued the token.
symbol	<a href="#">string</a>		The symbol of the transferred token.
amount	<a href="#">int64</a>		The amount of the transferred token.
memo	<a href="#">string</a>		The memo.

- **TotalTransactionFeesMap** The function of handling the transaction fees charged by ChargeTransactionFees. Output parameters: value, block\_hash, block\_height.

Field	Type	Label	Description
value	<a href="#">TotalTransactionFeesMap.ValueEntry</a>	repeated	Token dictionary that charge transaction fee, Symbol->Amount.
block_hash	<a href="#">Hash</a>		The hash of the block processing the transaction.
block_height	<a href="#">int64</a>		The height of the block processing the transaction.

- **TotalTransactionFeesMap.ValueEntry**

Field	Type	Label	Description
key	<a href="#">string</a>		
value	<a href="#">int64</a>		

- **TransactionFeeBill** The function of charging a transaction resource fee(Implemented from the ACS8 standards). Input parameters: fees\_map.

Field	Type	Label	Description
fees_map	<a href="#">TransactionFeeBill.FeesMapEntry</a>	repeated	The transaction fee dictionary, Symbol->fee.

- **TransactionFeeBill.FeesMapEntry**

Field	Type	Label	Description
key	<a href="#">string</a>		
value	<a href="#">int64</a>		

- **TransferFromInput** The function of transferring a specified amount of tokens from one address to another. Input parameters: from, to, symbol, amount, memo.

Field	Type	Label	Description
from	<a href="#">Address</a>		The source address of the token.
to	<a href="#">Address</a>		The destination address of the token.
symbol	<a href="#">string</a>		The symbol of the token to transfer.
amount	<a href="#">int64</a>		The amount to transfer.
memo	<a href="#">string</a>		The memo.



- **TransferInput** The function of transferring a given amount of tokens from one address to another. Input parameters: to, symbol, amount, memo.

Field	Type	Label	Description
to	<a href="#">Address</a>		The receiver of the token.
symbol	<a href="#">string</a>		The token symbol to transfer.
amount	<a href="#">int64</a>		The amount to to transfer.
memo	<a href="#">string</a>		The memo.

- **Transferred** The event of transferring tokens. Input parameters: from, to, symbol, amount, memo.

Field	Type	Label	Description
from	<a href="#">Address</a>		The source address of the transferred token.
to	<a href="#">Address</a>		The destination address of the transferred token.
symbol	<a href="#">string</a>		The symbol of the transferred token.
amount	<a href="#">int64</a>		The amount of the transferred token.
memo	<a href="#">string</a>		The memo.

- **TransferredBatch** The event of transferring the token. Input parameters: from, to, symbol, amount.

Field	Type	Label	Description
from	<a href="#">Address</a>		The source address of the transferred token.
to	<a href="#">Address</a>		The destination address of the transferred token.
symbol	<a href="#">string</a>	repeated	The symbol of the transferred token.
amount	<a href="#">int64</a>	repeated	The amount of the transferred token.

- **URI** The event of showing information of the changed token. Input parameters: symbol, amount.

Field	Type	Label	Description
-------	------	-------	-------------

Field	Type	Label	Description
symbol	<a href="#">string</a>		The symbol of issued token.
amount	<a href="#">int64</a>		The amount of issued token.

- **UnApproveInput** The function of reversing operation for Approve, it will decrease the allowance. Input parameters: spender, symbol, amount.

Field	Type	Label	Description
spender	<a href="#">Address</a>		The address that allowance will be decreased.
symbol	<a href="#">string</a>		The symbol of token to un-approve.
amount	<a href="#">int64</a>		The amount of token to un-approve.

- **UnApproved** The event of reversing operation for Approve, it will decrease the allowance. Input parameters: owner, spender, symbol, amount.

Field	Type	Label	Description
owner	<a href="#">Address</a>		The address of the token owner.
spender	<a href="#">Address</a>		The address that allowance be decreased.
symbol	<a href="#">string</a>		The symbol of un-approved token.
amount	<a href="#">int64</a>		The amount of un-approved token.

- **UnlockInput** The function of unlocking tokens. Input parameters: unlock\_address, lock\_id, symbol, amount, memo.

Field	Type	Label	Description
address	<a href="#">Address</a>		The one want to un-lock his token.
lock_id	<a href="#">Hash</a>		Id of the lock.
symbol	<a href="#">string</a>		The symbol of the token to un-lock.
memo	<a href="#">string</a>		a memo.
amount	<a href="#">int64</a>		The amount of tokens to un-lock.

- **UpdateCoefficientsInput** The function of updating the coefficient of the transaction fee calculation formula. Input parameters: piece\_numbers, coefficients.

Field	Type	Label	Description
piece_numbers	<a href="#">int32</a>	repeated	The specify pieces gonna update.
coefficients	<a href="#">CalculateFeeCoefficients</a>		Coefficients of one single type.

## Others

- **FeeTypeEnum**

Name	Number	Description
READ	0	
STORAGE	1	
WRITE	2	
TRAFFIC	3	
TX	4	

- **File-level Extensions**

Extension	Type	Base	Number	Description
is_indexed	bool	.google.protobuf.FieldOptions	502001	
identity	string	.google.protobuf.FileOptions	500001	
is_event	bool	.google.protobuf.MessageOptions	50100	
is_view	bool	.google.protobuf.MethodOptions	506001	
base	string	.google.protobuf.ServiceOptions	505001	
csharp_state	string	.google.protobuf.ServiceOptions	505030	

- **Scalar Value Types**

.proto Type	Notes	C++	Java	Python	Go	C#	PHP	Ruby
<a href="#">double</a>		double	double	float	float64	double	float	Float
<a href="#">float</a>		float	float	float	float32	float	float	Float

.proto Type	Notes	C++	Java	Python	Go	C#	PHP	Ruby
<a href="#">int32</a>	Uses variable-length encoding. Inefficient for encoding negative numbers – if your field is likely to have negative values, use sint32 instead.	int32	int	int	int32	int	integer	Bignum or Fixnum (as required)
<a href="#">int64</a>	Uses variable-length encoding. Inefficient for encoding negative numbers – if your field is likely to have negative values, use sint64 instead.	int64	long	int/long	int64	long	integer/string	Bignum
<a href="#">uint32</a>	Uses variable-length encoding.	uint32	int	int/long	uint32	uint	integer	Bignum or Fixnum (as required)
<a href="#">uint64</a>	Uses variable-length encoding.	uint64	long	int/long	uint64	ulong	integer/string	Bignum or Fixnum (as required)
<a href="#">sint32</a>	Uses variable-length encoding. Signed int value. These more efficiently encode negative numbers than regular int32s.	int32	int	int	int32	int	integer	Bignum or Fixnum (as required)
<a href="#">sint64</a>	Uses variable-length encoding. Signed int value. These more efficiently encode negative numbers than regular int64s.	int64	long	int/long	int64	long	integer/string	Bignum

.proto Type	Notes	C++	Java	Python	Go	C#	PHP	Ruby
<a href="#">fixed32</a>	Always four bytes. More efficient than uint32 if values are often greater than 2 <sup>28</sup> .	uint32	int	int	uint32	uint	integer	Bignum or Fixnum (as required)
<a href="#">fixed64</a>	Always eight bytes. More efficient than uint64 if values are often greater than 2 <sup>56</sup> .	uint64	long	int/long	uint64	ulong	integer/string	Bignum
<a href="#">sfixed32</a>	Always four bytes.	int32	int	int	int32	int	integer	Bignum or Fixnum (as required)
<a href="#">sfixed64</a>	Always eight bytes.	int64	long	int/long	int64	long	integer/string	Bignum
<a href="#">bool</a>		bool	boolean	boolean	bool	bool	boolean	TrueClass/FalseClass
<a href="#">string</a>	A string must always contain UTF-8 encoded or 7-bit ASCII text.	string	String	str/unicode	string	string	string	String (UTF-8)
<a href="#">bytes</a>	May contain any arbitrary sequence of bytes.	string	ByteString	str	[]byte	ByteString	string	String (ASCII-8BIT)

## Bibliography

Bibliographical references are resources that provide additional or helpful material but do not need to be understood or used to implement this standard. Reference to these resources is made for informational use only.

[B1] ISO 22739:2020(en), Blockchain and distributed ledger technologies—Vocabulary.

[B2] ERC-20 Token Standard.

[B3] ERC-721 Token Standard.