

VolumeRestrictionTransferManager

- Introduced in: 2.1.0
- Contract name: VolumeRestrictionTM
- Type: Transfer Manager Module
- State machine: <https://www.lucidchart.com/documents/edit/ab80f6a1-c403-402e-9dc0-f4539c96668b/0?shared=true&>

How it works

It is used to restrict the maximum volume of tokens being traded by any individual investor in a given period of time and it also restricts maximum transfers that should be across all the token holders associated. Issuer can also exempt any token holders by adding them into the exemption list to make them unaffected by this module.

Key functionalities (as defined in the Smart Contract)

Initialization

This module is initialized with no parameters. That means during creation of this contract there's no need to call any type of `configure()` function.

Using the module

Layout for the Individual restriction

Note - Below use-case only works when the Alice have only `Individual restriction` not `Individual daily restriction (24 hrs rolling period)`.

UseCase : Issuer wants to restrict Alice's transfers as per the volume of tokens. Issuer will do so with the following transaction.

Call the `addIndividualRestriction()` function with following data.

address of Alice → 0xabc..

allowedTokens → 10,000 (If `typeOfRestriction` is `Percentage` then

allowedTokens will be in the % (w.r.t to totalSupply) with a multiplier of 10^{16} .

else it will be fixed amount of tokens with a multiplier of 10^{18})
startTime → now
rollingPeriodInDays → 5
endTime → now + 12 days
restrictionType → 0 . (Fixed restriction)

Now Alice starts trading with following assumptions.

Alice's Address != 0x0 , paused = false , exemptionList[0xabc] = false

Day1: Alice tries to sell 1000 tokens

amount = 1000

sumOfLastPeriod = 0

$10000 \geq 1000 + 0$ ---- yes transaction processed sumOfLastPeriod = 1000

Day 2: Alice tries to sell 5000 tokens

amount = 5000

sumOfLastPeriod = 1000

$10000 \geq 5000 + 1000$. ----- yes tx processed sumOfLastPeriod = 6000

Day3 : No trading

Day4: No trading

Day5 : Alice tries to sell 6000 tokens

amount = 6000

sumOfLastPeriod= 6000

$10000 \geq 6000 + 6000$ ---- false tx failed sumOfLastPeriod = 6000

----- First rolling period ends -----

Day6: Alice tries to sell 3000 tokens

amount = 3000

sumOfLastPeriod = 6000 - 1000(amount sold on day 1) => 5000

$10000 \geq 5000 + 3000$ ---- true tx processed new sumOfLastPeriod = 8000

Day7: No trading

Day8: Alice tries to sell 4000 tokens

amount = 4000

sumOfLastPeriod = 8000 - 5000(amount sold on day 2) - 0 (amount sold on day 3) => 3000 or (Day 4, 5, 6, 7, 8 trade sum which is 3000).

$10000 \geq 4000 + 3000$ — yes tx processed sumOfLastPeriod = 7000

continues

Transfer Verification

- If `_from` address is 0x0 or present in the exemption list or `paused` variable is true then transfers will be unaffected by this module.
- If `_from` address has any individual restriction, whether `Individual` or `Individual daily` or both. `isDefault` will be false in `_restrictionCheck()` function params. Otherwise it will be true.
- 4 cases related to `_from` address
 - `_from` has Individual Restriction
 - `_from` has 24 hrs Individual daily restriction
 - `_from` has both individual restriction and individual daily restriction
 - `_from` doesn't have any Individual restriction. It falls into the default restriction. Whatever the default restriction will be active that time will apply, whether `default` or `default daily` or both.

First Case :-

- If `_from` has Individual restriction only & transaction time is between `startTime` and `endTime` of the restriction, then transaction will go through only when:
 - `_amount` is less than or equal to the `_allowedAmount - sumOfLastPeriod`, where `_allowedAmount` is the fixed number of tokens (`allowedTokens`) allowed to transact in a given rolling period when restriction type is `Fixed`. If not, then `allowedAmount` will be calculated at the tx processing time according to the current totalSupply of the ST. i.e `_allowedAmount = (_restriction.allowedTokens.mul(ISecurityToken(securityToken).totalSupply())) / uint256(10) ** 18`. `sumOfLastPeriod` will be the sum of the volume traded by the `_from` in last `n` days, where `n` is always less than or equal to the `rollingPeriod`. (given `n` will always be calculated after the `startTime` of the individual restriction).

Second Case :-

- If `_from` address has individual daily restriction & transaction time is between `startTime` and `endTime` of the restriction, then transaction will go through only when:
 - `_amount` is less than or equal to the `_allowedAmount - txSumOfDay`, where `_allowedAmount` will be calculated same as above but according to the

individual daily volume restriction. While `txSumOfDay` is the amount traded till now within the 24 hrs span*. This amount is stored according at the start time of 24 hrs i.e `dailyLastTradedDayTime`.

Third Case :-

- If `_from` address has (individual daily restriction + individual restriction) & transaction time is between `startTime` and `endTime` of the restriction, then transaction will go through only when:
 - `_amount` is less than or equal to the `_allowedAmount - sumOfLastPeriod` (First case)
 - `_amount` is less than or equal to the `_allowedAmount - txSumOfDay` (Second Case)

Both above condition should be true otherwise tx gets failed. #807.

Fourth Case :-

- It works similar to any one of the first/second/third case. Depends upon which default restriction is active. The only difference between above three and this is to use different storage variables to store the data.

Add Volume restriction

For Individual restriction for a given holder address.

```
/**
 * @notice Use to add the new individual restriction for a
 * given token holder
 * @param _holder Address of the token holder, whom restriction
 * will be implied
 * @param _allowedTokens Amount of tokens allowed to be trade
 * for a given address.
 * @param _startTime Unix timestamp at which restriction get
 * into effect
 * @param _rollingPeriodInDays Rolling period in days (Minimum
 * value should be 1 day)
 * @param _endTime Unix timestamp at which restriction effects
 * will gets end.
```

```

    * @param _restrictionType Whether it will be `Fixed` (fix
ed no. of tokens allowed to transact)
    * or `Percentage` (tokens are calculated as per the total
Supply in the fly).
    */
function addIndividualRestriction(
    address _holder,
    uint256 _allowedTokens,
    uint256 _startTime,
    uint256 _rollingPeriodInDays,
    uint256 _endTime,
    RestrictionType _restrictionType
)
    public
    withPerm(ADMIN)

```

Require checks

- `individualRestriction[_holder].endTime < now` it means if holder has no restriction or holder already has restriction but it is expired then it will allow the add new restriction otherwise tx get reverted.
- `_holder` should not be a present in the exemption list.
- `_restrictionType` could be 0 or 1. No other value allowed.
- `_rollingPeriodInDays` always between [1, 365].
- `_restrictionType == 0` then `_allowedTokens` always greater then zero otherwise `_allowedTokens` should be non zero and value lies between (0, $100 * 10^{16}$].
- Difference of days between `_startTime` and `_endTime` should be \geq `_rollingPeriodInDays`
- If `_startTime` is 0 then it will takes current block timestamp + 1 as the startTime.

For multiple `_holders` (Similar require checks are applied to it) :

```
/**
```

```

        * @notice Use to add the new individual restriction for multiple token holders

        * @param _holders Array of address of the token holders, whom restriction will be implied

        * @param _allowedTokens Array of amount of tokens allowed to be trade for a given address.

        * @param _startTimes Array of unix timestamps at which restrictions get into effect

        * @param _rollingPeriodInDays Array of rolling period in days (Minimum value should be 1 day)

        * @param _endTimes Array of unix timestamps at which restriction effects will gets end.

        * @param _restrictionTypes Array of restriction types value whether it will be `Fixed` (fixed no. of tokens allowed to transact)

        * or `Percentage` (tokens are calculated as per the total Supply in the fly).
    */
    function addIndividualRestrictionMulti(
        address[] memory _holders,
        uint256[] memory _allowedTokens,
        uint256[] memory _startTimes,
        uint256[] memory _rollingPeriodInDays,
        uint256[] memory _endTimes,
        RestrictionType[] memory _restrictionTypes
    )
        public

```

For Individual daily restriction:

```

/**
    * @notice Use to add the new individual daily restriction for all token holder

```

```

        * @param _holder Address of the token holder, whom restriction will be implied

        * @param _allowedTokens Amount of tokens allowed to be traded for all token holder.

        * @param _startTime Unix timestamp at which restriction get into effect

        * @param _endTime Unix timestamp at which restriction effects will gets end.

        * @param _restrictionType Whether it will be `Fixed` (fixed no. of tokens allowed to transact)

        * or `Percentage` (tokens are calculated as per the total Supply in the fly).

    */
    function addIndividualDailyRestriction(
        address _holder,
        uint256 _allowedTokens,
        uint256 _startTime,
        uint256 _endTime,
        RestrictionType _restrictionType
    )
        public
        withPerm(ADMIN)

```

In Individual daily restriction `_rollingPeriodInDays` always hardcoded to `1`.

Individual daily restriction for multiple holders:

```

/**
    * @notice Use to add the new individual daily restriction for multiple token holders

    * @param _holders Array of address of the token holders, whom restriction will be implied

    * @param _allowedTokens Array of amount of tokens allowed to be trade for a given address.

```

```

    * @param _startTimes Array of unix timestamps at which re
strictions get into effect

    * @param _endTimes Array of unix timestamps at which rest
riction effects will gets end.

    * @param _restrictionTypes Array of restriction types val
ue whether it will be `Fixed` (fixed no. of tokens allowed to
transact)

    * or `Percentage` (tokens are calculated as per the total
Supply in the fly).
*/
function addIndividualDailyRestrictionMulti(
    address[] memory _holders,
    uint256[] memory _allowedTokens,
    uint256[] memory _startTimes,
    uint256[] memory _endTimes,
    RestrictionType[] memory _restrictionTypes
)
    public

```

For Global restriction(same require check applies):

```

/**
    * @notice Use to add the new default restriction for all
token holder

    * @param _allowedTokens Amount of tokens allowed to be tr
aded for all token holder.

    * @param _startTime Unix timestamp at which restriction g
et into effect

    * @param _rollingPeriodInDays Rolling period in days (Min
imum value should be 1 day)

    * @param _endTime Unix timestamp at which restriction eff
ects will gets end.

```



```

    * @param _restrictionType Whether it will be `Fixed` (fixed no. of tokens allowed to transact)
    * or `Percentage` (tokens are calculated as per the total Supply in the fly).
    */
function addDefaultRestriction(
    uint256 _allowedTokens,
    uint256 _startTime,
    uint256 _rollingPeriodInDays,
    uint256 _endTime,
    RestrictionType _restrictionType
)
    external
    withPerm(ADMIN)

```

For default daily restriction:

```

/**
    * @notice Use to add the new default daily restriction for all token holder
    * @param _allowedTokens Amount of tokens allowed to be traded for all token holder.
    * @param _startTime Unix timestamp at which restriction gets into effect
    * @param _endTime Unix timestamp at which restriction effects will get end.
    * @param _restrictionType Whether it will be `Fixed` (fixed no. of tokens allowed to transact)
    * or `Percentage` (tokens are calculated as per the total Supply in the fly).
    */
function addDefaultDailyRestriction(
    uint256 _allowedTokens,

```

```

        uint256 _startTime,
        uint256 _endTime,
        RestrictionType _restrictionType
    )
    external
    withPerm(ADMIN)

```

same require checks only one change that `_rollingPeriodInDays` values is by default 1.

Modify Volume Restrictions

For modifying the individual restriction**

```

/**
 * @notice Use to modify the existing individual restricti
on for a given token holder
 * @param _holder Address of the token holder, whom restri
ction will be implied
 * @param _allowedTokens Amount of tokens allowed to be tr
ade for a given address.
 * @param _startTime Unix timestamp at which restriction g
et into effect
 * @param _rollingPeriodInDays Rolling period in days (Min
imum value should be 1 day)
 * @param _endTime Unix timestamp at which restriction eff
ects will gets end.
 * @param _restrictionType Whether it will be `Fixed` (fix
ed no. of tokens allowed to transact)
 * or `Percentage` (tokens are calculated as per the total
Supply in the fly).
 */
function modifyIndividualRestriction(
    address _holder,
    uint256 _allowedTokens,

```

```

        uint256 _startTime,
        uint256 _rollingPeriodInDays,
        uint256 _endTime,
        RestrictionType _restrictionType
    )

    public
    withPerm(ADMIN)

```

It follows the same require checks as `addxxx..` functions have.

For modifying multiple individual restrictions:

```

function modifyIndividualRestrictionMulti(
    address[] _holders,
    uint256[] _allowedTokens,
    uint256[] _startTimes,
    uint256[] _rollingPeriodInDays,
    uint256[] _endTimes,
    RestrictionType[] _restrictionTypes
)

    public
    withPerm(ADMIN)

{

```

For modifying the individual daily restriction

```

/**
    * @notice Use to modify the existing individual daily res
    triction for a given token holder
    * @dev Changing of startTime will affect the 24 hrs span.
    i.e if in earlier restriction days start with
    * morning and end on midnight while after the change day
    may start with afternoon and end with other day afternoon

```

```

    * @param _holder Address of the token holder, whom restriction will be implied

    * @param _allowedTokens Amount of tokens allowed to be trade for a given address.

    * @param _startTime Unix timestamp at which restriction gets into effect

    * @param _endTime Unix timestamp at which restriction effects will gets end.

    * @param _restrictionType Whether it will be `Fixed` (fixed no. of tokens allowed to transact)

    * or `Percentage` (tokens are calculated as per the total Supply in the fly).

    */
function modifyIndividualDailyRestriction(
    address _holder,
    uint256 _allowedTokens,
    uint256 _startTime,
    uint256 _endTime,
    RestrictionType _restrictionType
)
    public
    withPerm(ADMIN)

```

Modifying the Multiple daily restrictions

```

function modifyIndividualDailyRestrictionMulti(
    address[] _holders,
    uint256[] _allowedTokens,
    uint256[] _startTimes,
    uint256[] _endTimes,
    RestrictionType[] _restrictionTypes
)
    public

```

```
        withPerm(ADMIN)
    {
```

Modifying default restriction:

```
/**
 * @notice Use to modify the global restriction for all to
ken holder
 * @param _allowedTokens Amount of tokens allowed to be tr
aded for all token holder.
 * @param _startTime Unix timestamp at which restriction g
et into effect
 * @param _rollingPeriodInDays Rolling period in days (Min
imum value should be 1 day)
 * @param _endTime Unix timestamp at which restriction eff
ects will gets end.
 * @param _restrictionType Whether it will be `Fixed` (fix
ed no. of tokens allowed to transact)
 * or `Percentage` (tokens are calculated as per the total
Supply in the fly).
 */
function modifyDefaultRestriction(
    uint256 _allowedTokens,
    uint256 _startTime,
    uint256 _rollingPeriodInDays,
    uint256 _endTime,
    RestrictionType _restrictionType
)
    external
    withPerm(ADMIN)
```

Modifying daily default restriction

```
/**
```

```

    * @notice Use to modify the daily default restriction for
    all token holder

    * @dev Changing of startTime will affect the 24 hrs span.
    i.e if in earlier restriction days start with

    * morning and end on midnight while after the change day
    may start with afternoon and end with other day afternoon.

    * @param _allowedTokens Amount of tokens allowed to be tr
    aded for all token holder.

    * @param _startTime Unix timestamp at which restriction g
    et into effect

    * @param _endTime Unix timestamp at which restriction eff
    ects will gets end.

    * @param _restrictionType Whether it will be `Fixed` (fix
    ed no. of tokens allowed to transact)

    * or `Percentage` (tokens are calculated as per the total
    Supply in the fly).

    */
function modifyDefaultDailyRestriction(
    uint256 _allowedTokens,
    uint256 _startTime,
    uint256 _endTime,
    RestrictionType _restrictionType
)
    external
    withPerm(ADMIN)

```

Remove Volume Restriction

Similar to addition or modification contract has the `removeXX()` or `removeXXMulti()` functions that will use to remove the individual, individual daily default, default daily restriction.

Getters

Provide the Individual bucket details for a given address.

```

/**
 * @notice Use to get the bucket details for a given address
 *
 * @param _user Address of the token holder for whom the bucket details has queried
 * @return uint256 lastTradedDayTime
 * @return uint256 sumOfLastPeriod
 * @return uint256 days covered
 * @return uint256 24h lastTradedDayTime
 * @return uint256 Timestamp at which last transaction get executed
 */
function getIndividualBucketDetailsToUser(address _user) public view returns(uint256, uint256, uint256, uint256, uint256)

```

Provide the Default bucket details for a given address.

```

/**
 * @notice Use to get the bucket details for a given address
 *
 * @param _user Address of the token holder for whom the bucket details has queried
 * @return uint256 lastTradedDayTime
 * @return uint256 sumOfLastPeriod
 * @return uint256 days covered
 * @return uint256 24h lastTradedDayTime
 * @return uint256 Timestamp at which last transaction get executed
 */
function getDefaultBucketDetailsToUser(address _user) public view returns(uint256, uint256, uint256, uint256, uint256)

```

Use to get the volume of token that being traded at a particular day for a given address

```
/**
 * @notice Use to get the volume of token that being trade
d at a particular day (`_at` + 24 hours) for a given user
 * @param _user Address of the token holder
 * @param _at Timestamp
 */
function getTotalTradedByUser(address _user, uint256 _at)
external view returns(uint256)
```

Use to get the balance of the token holder for the given partition

```
/**
 * @notice return the amount of tokens for a given user as
per the partition
 * @param _partition Identifier
 * @param _tokenHolder Whom token amount need to query
 * @param _additionalBalance It is the `_value` that trans
fer during transfer/transferFrom function call
 */
function getTokensByPartition(bytes32 _partition, address
_tokenHolder, uint256 _additionalBalance) external view return
s(uint256)
```

- `getExemptAddress()` use to return the list of exempted addresses.
- `getIndividualRestriction(address _investor)` use to return the individual restriction details for a given user.
- `getIndividualDailyRestriction(address _investor)` use to return the individual daily restriction details for a given user.
- `getDefaultRestriction()` use to return the default restriction details.
- `getDefaultDailyRestriction()` use to return the default daily restriction details.

- `getRestrictionData()` Provide the restriction details of all the restricted addresses.

Other Functions

Below function will be used to add/remove the wallet address from the exemption list.

```
/**
 * @notice Add/Remove wallet address from the exempt list
 * @param _wallet Ethereum wallet/contract address that ne
ed to be exempted
 * @param _change Boolean value used to add (i.e true) or
remove (i.e false) from the list
 */
function changeExemptWalletList(address _wallet, bool _cha
nge) public withPerm(ADMIN)
```

Special considerations / notes

** modification only allowed when the former restriction `startTime` should be greater than the current timestamp.

* 24hrs span will not be always morning to midnight of a day. It can be afternoon of day to the afternoon of the other day. It always depends on the `startTime` of the Daily restriction.

Gas Estimations (Old)

Estimated gas for addIndividualRestriction:	162557
Gas estimation (Individual Restriction, First time trade):	144358
Gas Estimation for the Individual daily tx (First time trade):	124819
Gas Estimation for the Individual daily tx (Tx on same day) -	89616
Gas Estimation for the Individual daily tx (Next day starting) -	109936
Gas estimation (Individual + Individual daily tx):	178954
Gas estimation (Default Daily):	126300
Gas estimation (Default):	144909
Gas estimation (Default + default daily):	162598

Troubleshooting / FAQs

None

Know Issues / bugs

All batch functions will hit the block gas limits if the array length is greater than 80(approx.).