

General Permission Manager

- **Introduced in:** 1.3.0
- **Last updated:** 2.0.0
- **Contract name:** GeneralPermissionManager.sol
- **Compatible ST Protocol version range:** TBD
- **Type:** Permission Manager Module

How it works

The GPM allows the issuer to add wallets as delegates and to give them permissions to use restricted functions on other modules. Every time one of these functions is called on a module, it will check if the sender is allowed to do it.

For example, the issuer can give an associate the ability to work on the tokens' whitelist by giving his address the `WHITELIST` permission on the `GeneralTransferManager`.

Key functionalities (as defined in the Smart Contract)

Initialization

This module is initialized without any parameters.

Checking permissions

Called by modules when restricted functions are called by a non-owner wallet.

If the address passed is a delegate and it has the required permission for the given module and function, it returns true and it allows to execute the restricted function on that module.

Otherwise it returns false.

```
function checkPermission(address _delegate, address _module, bytes32 _perm) external view returns(bool)
```

Adding delegates

To change a permission, it is necessary to have added a delegate previously. This can be done by calling the following function, where `_details` is a required description (I.E: "Business partner", "KYC Partner") for the `_delegate`.

```
function addDelegate(address _delegate, bytes32 _details) external  
withPerm(CHANGE_PERMISSION)
```

This means the right operations order is:

1. Add a delegate
2. Add as many permissions as needed on different modules to an existing delegate.

Only issuer or a designated ADMIN (through this module) can add delegates.

Changing permissions

Once delegates are added, this module can change the permissions assigned to them (allow or disallow).

- Change a single permission for a single module for a delegate by calling:

```
function changePermission(address _delegate, address _module,  
bytes32 _perm, bool _valid) public withPerm(CHANGE_PERMISSION)
```

- Change multiple permissions for multiple modules for a delegate by calling:

```
function changePermission(address _delegate, address[] _modules,  
bytes32[] _perms, bool[] _valids) public withPerm(CHANGE_PERMISSION)
```

Only issuer or a designated ADMIN (through this module) can change permissions.

Removing delegates

By removing a delegate, all its permission will be removed too. This means that if the delegate is added again in the future, it will be necessary to change its permissions again.

This can be done by calling: `function deleteDelegate(address _delegate)`
`external withPerm(CHANGE_PERMISSION).`

Only issuer or a designated ADMIN (through this module) can change permissions.

Reading data stored

This module provides several functions to access delegates and permissions:

- Check if an address is a delegate or not by calling: `function`
`checkDelegate(address _potentialDelegate) external view`

returns(bool)

- Get all delegates with a given permission and module by calling: `function getAllDelegatesWithPerm(address _module, bytes32 _perm) external view returns(address[])` - **Should be called off-chain only**
- Get all permissions for a delegate through all modules of the given types for a security token by calling: `getAllModulesAndPermsFromTypes(address _delegate, uint8[] _types, address _tokenAddress) external view returns(address[], bytes32[])` - **Should be called off-chain only**
- Get all delegates by calling: `function getAllDelegates() external view returns(address[])`

Special considerations / notes

None

Troubleshooting / FAQs

None

Known Issues / bugs

None