

# asdOFT

[asdOFT](#) is the standard contract for all asD tokens. It utilizes LayerZero's Omnichain Fungible Token (OFT) standard to enable bridging across supported networks.

## Deployment

In order to deploy an asD token, simply deploy a new instance of asdOFT on Canto. In the constructor, specify:

- `_name`
  - – the name of your asD token
- `_symbol`
  - – the symbol of your asD token
- `_lzEndpoint`
  - – the contract address for the LayerZero endpoint on Canto, i.e. `0x1a44076050125825900e736c501f859c50fe728c`
- `_cNote`
  - – the contract address for [cNOTE](#)
    - i.e. `0xEe602429Ef7eCe0a13e4FfE8dBC16e101049504C`
- `_csrRecipient`
  - – the address to register for [Contract Secured Revenue](#)
    - e.g. your wallet
- 

## Methods

### mint

The `mint(uint256 _amount)` method allows users to permissionlessly mint an asD token by depositing NOTE, which is automatically supplied to the Canto Lending Market.

Before calling this method, a user must first approve the contract to spend their NOTE.

### burn

The `burn(uint256 _amount)` method allows holders of an asD token to burn the asD token, thereby withdrawing the underlying NOTE from the Canto Lending Market and returning it to the user.

### withdrawCarry

The `withdrawCarry(uint256 _amount)` method allows the owner of an asD token to withdraw carry (accrued interest) earned on NOTE backing that has been supplied to the Canto Lending Market. It ensures that a 1:1 NOTE:asD exchange rate is maintained after withdrawal.

[Previous Application Specific Dollar](#) [Next asdRouter](#) Last updated 4 days ago On this page \* [Deployment](#) \* [Methods](#) \* [mint](#) \* [burn](#) \* [withdrawCarry](#)