

Wrapped FIL

Wrapped FIL (wFIL) is the canonical wrapper token of the native Filecoin (FIL) token. Wrapped FIL features a 1-to-1 ratio pegged to FIL.

Wrapped FIL (wFIL) is a wrapper token based on the ERC-20 token standard for the native Filecoin token (FIL). It allows FIL to be bridged and used in Ethereum-compatible decentralized applications (dapps) hosted on other blockchains, such as decentralized exchanges (DEXs), lending platforms, and other places where FIL is not natively supported.

Wrapped FIL operates like any other ERC20-wrapped native blockchain token: a user deposits FIL into the wFIL contract and gets back an equal number of wFIL tokens. When users want to convert their wFIL back to FIL, they can burn the wFIL and unlock the same amount of FIL that was initially locked in the wFIL contract.

Overall, wFIL provides additional liquidity and interoperability for FIL tokens, making the Filecoin network more accessible for a broader range of decentralized finance (defi) use cases across multiple blockchains.

When wrapping and unwrapping FIL ensure you are using the correct wFIL contract address on Filecoin.

Wrapped FIL contract addresses

Only use the following addresses when wrapping and unwrapping FIL:

- Mainnet:0x60E1773636CF5E4A227d9AC24F20fEca034ee25A
- Calibration testnet:0xaC26a4Ab9cF2A8c5DBaB6fb4351ec0F4b07356c4
-

Wrapping and unwrapping process

There are a couple of options for users to wrap and unwrap FIL using a web browser:

- [Glif](#)
- [wfil.io](#)
-

To wrap FIL into wFIL, follow these steps:

1. Obtain FIL
2. : Ensure you have FIL in your MetaMask wallet before wrapping it.
3. Connect your wallet
4. : You will need to connect your wallet to a platform that supports wFIL wrapping, such as [Glif](#)
5. or [wfil.io](#)
6. .
7. Wrap your FIL
8. : After you've connected your wallet, you can wrap your FIL by following the platform's instructions. Generally, you'll need to select the amount of FIL you want to wrap and confirm the transaction on MetaMask. The platform will then mint an equivalent amount of wFIL and deposit it into your wallet.
9. Use wFIL
10. : Once you have wFIL in your wallet, you can use it on various Defi products that support token swapping or bridging wFIL to other blockchains.
- 11.

To unwrap FIL and receive FIL back to your wallet, users can directly go to supported platforms such as [Glif](#) or [wfil.io](#) to unwrap FIL following the platform's instructions. Once the network confirms the unwrap transaction, FIL tokens are transferred back to your wallet address.

Programmatic interaction

Developers integrating wFIL into applications or protocols can wrap and unwrap FIL programmatically. The wFIL smart contract is deployed on the Filecoin Mainnet and Calibration testnet.

Wrap FIL

To add wrapping features to a project, developers must interact with the wFIL smart contract that manages the wFIL minting and burning process. The source code of the wFIL smart contract is in the [wFIL GitHub repo](#) .

Do not directly send FIL to the wFIL contract address. Also, ensure you do not send FIL using the `METHOD_SEND` method. Always use the `invokeEVM` method.

There are two options to wrap FIL:

1. Call the deposit()
2. method in the wFIL contract and attach the amount of FIL tokens users want to wrap. This process will mint wFIL 1:1 and transfer to the msg.sender
3. address.
- 4.

...

```
Copy function deposit() public payable virtual {
    _mint(msg.sender, msg.value);
    emit Deposit(msg.sender, msg.value);
}
```

...

1. Since the wFIL implements the receive function, you can send FIL to the wFIL contract using the `invokeEVM`
2. method to wrap FIL. This method will trigger the `deposit`
3. function, minting the caller with wFIL 1:1.
- 4.

...

```
Copy receive() external payable virtual {
    deposit();
}
```

...

Unwrap FIL

To unwrap wFIL into FIL, developers need to call the `withdraw` method in the wFIL contract and specify how many wFIL you would like to unwrap. The `withdraw` method looks like this:

...

```
Copy function withdraw(uint _amount) public virtual {
    _burn(msg.sender, _amount);
    emit Withdrawal(msg.sender, _amount);
    payable(msg.sender).sendValue(_amount);
}
```

...

This process will burn the amount of wFIL from the caller's balance and transfer the unwrapped FIL 1:1 back to the caller's address.

[Previous](#) [Advanced](#) [Next](#) [Oracles](#)

Last updated 2 months ago