

## EIP 777: ERC-777 Token Standard

### [Return to Ethereum ERCs](#)

**Note:** The following is an excerpt from the official Ethereum site. It is provided here as a convenience and is not authoritative. Refer to the original document as the authoritative reference.

Table 1: Data sheet for Token Standard

Title	Token Standard
Author	Jacques Dafflon, Jordi Baylina, Thomas Shababi
Status	Final
Created	2017-11-20
Description	<a href="http://eips.ethereum.org/EIPS/eip-777">http://eips.ethereum.org/EIPS/eip-777</a>
Specification	<a href="http://eips.ethereum.org/EIPS/eip-777#Specification">http://eips.ethereum.org/EIPS/eip-777#Specification</a>
Category	ERC
Requires	<a href="#">1820</a>

### Abstract

*This standard defines a new way to interact with a token contract while remaining backward compatible with ERC20.*

*It defines advanced features to interact with tokens. Namely, operators to send tokens on behalf of another address—contract or regular account—and send/receive hooks to offer token holders more control over their tokens.*

*It takes advantage of [ERC1820](#) to find out whether and where to notify contracts and regular addresses when they receive tokens as well as to allow compatibility with already-deployed contracts.*

### Motivation

*This standard tries to improve upon the widely used ERC20 token standard. The main advantages of this standard are:*

- *Uses the same philosophy as [Ether](#) in that [tokens](#) are sent with `send(dest, value, data)`.*
- *Both contracts and regular addresses can control and reject which token they send by registering a `tokensToSend` hook. (Rejection is done by reverting in the hook function.)*
- *Both contracts and regular addresses can control and reject which token they receive by registering a `tokensReceived` hook. (Rejection is done by reverting in the hook function.)*
- *The `tokensReceived` hook allows to send tokens to a contract and notify it in a single transaction, unlike ERC20 which requires a double call (`approve/transferFrom`) to achieve this.*

- *The holder can “authorize” and “revoke” operators which can send tokens on their behalf. These operators are intended to be verified contracts such as an exchange, a cheque [processor](#) or an automatic charging system.*
- *Every token transaction contains data and operatorData bytes fields to be used freely to pass data from the holder and the operator, respectively.*

It is backward compatible with [wallets](#) that do not contain the tokensReceived hook function by deploying a proxy contract implementing the tokensReceived hook for the wallet.

From:  
<https://www.omgwiki.org/dido/> - **DIDO Wiki**

Permanent link:  
[https://www.omgwiki.org/dido/doku.php?id=dido:public:ra:xapend:xapend.b\\_stds:defact:ethereum:eip:erc\\_0777](https://www.omgwiki.org/dido/doku.php?id=dido:public:ra:xapend:xapend.b_stds:defact:ethereum:eip:erc_0777)

Last update: **2021/08/18 11:18**

