Edit on Github

Assets

English

With Counterparty, users can create their own assets (also known as “tokens”, “coins” or “currencies”) *inside* the Bitcoin blockchain. These are seperate from Bitcoin the currency itself, but exist entirely inside ordinary Bitcoin transactions. Tokens can be received, stored, and sent from any Bitcoin address to any other. They can also be placed in cold storage. Unlike Colored Coins, Counterparty tokens are *not* tied to the BTC balance of any given address. This means that sending/receiving bitcoins has no effect on the balance of tokens.

Among other features, Counterparty adds the ability *create*, *send*, *trade*, and *pay distributions on* assets, in a fully decentralized and trustless manner. While Counterparty has its own internal currency (XCP), trading and creating assets does not require anything apart from regular Bitcoin transaction fees.

Many of the features described below can be accessed using the Web-based Counterwallet. Especially casual users and those without a `counterparty-cli` setup can benefit from the convenience of Counterwallet.

Creating assets

Counterparty allows users to *issue assets*. An asset that is created within the Counterparty protocol is often called a *user-created token*. User-created tokens are just as real as XCP or even BTC. With the asset issuance function, every user has the ability to create a new currency project inside the Bitcoin and Counterparty ecosystem.

You can create two different types of assets:

1. **Named**: A unique string of 4 to 12 uppercase Latin characters (inclusive) not beginning with ‘A’. Alphabetic tokens carry a one-time issuance fee of `0.5 XCP` to discourage spam and squatting. This fee is burned (permanently taken out of circulation). `BTC` and `XCP` are the only three-character asset names. For more information, see the Assets section in the Counterparty specification.

2. **Numeric (Free)**: An integer between `26^12 + 1` and `256^8` (inclusive), prefixed with `A`. Numeric assets only require one Bitcoin transaction fee to be created.

The different kinds of assets

English

The most basic kind of asset must specify:

- who is issuing it (`source`)
- the name of the asset (`asset`)
- how much of `asset` is being issued (`quantity`)
- a description of asset (`description`)

It is possible to issue more of `asset` , but, at any one time, there can only be one address which issues `asset` . With that said, the Counterparty protocol allows `source` to transfer issuance rights of `asset` . Moreover, an asset can also be locked, so that there can be no further issuances of it. (See the examples for instructions on how to do this with `counterparty-cli`). A description must always be included, even if `description` is just an empty string; the syntax of an asset *with no description* is `description=""` .

Beyond creating the most basic asset, it is also possible to make assets either *divisible* or *callable*. If an asset is made divisible (or callable) upon its initial issuance, it must always be divisible (or callable) with every issuance thereafter. A divisible user-created asset is, like, Bitcoin and XCP, divisible up to 8 decimal places. A callable asset is an asset which the issuer can call back (i.e. repurchase) from its owners at a date (`call-date`) and for a price (`call-price`) specified at the initial issuance.

Sending assets (`send`)

To send an asset in Counterparty, one must specify:

- who is sending the asset (`source`)
- what asset `source` is sending (`asset`)
- how much of `asset` `source` is sending (`quantity`)
- to whom `source` is sending `quantity` of asset (`destination`)

Paying distributions on assets

It is possible to distribute funds proportionally among asset holders using the `distribution` function. This feature is also also known as `dividend payments` , depending on their desired [illegible]butions are paid in in any `distribution_asset` to everyone who holds the

English

asset in proportion to how many units he holds; specifically: Let `total` equal the total distribution paid out, and `quantity` be the total amount of asset, then: `quantity-per-unit = total/quantity`

Distributions can be paid out to any assets that you ownership and control over. You can freely select the currency in which distributions are to be paid out: BTC, XCP, or any other user-created asset.

Trading on the decentralized exchange

Counterparty supports *peer-to-peer asset exchange*: users can trade assets with no middleman and no counterparty risk. The platform upon which trading is done is Counterparty's *decentralized exchange* and the Bitcoin blockchain. In what follows trading on the decentralized exchange will be detailed and explained by means of examples. For the purposes of the following use-cases:

- "ordern" denotes the *nth* order in time, `give_asset n` denotes the asset being given in the order, etc.
- Sally's creates order1 and Alice creates order2
- `give_asset2 = get_asset1`

Creating an order

At its most basic level, a trade on Counterparty's decentralized exchange consists of two *orders*, which are *matched* by the protocol. When Sally is constructing her order, she must specify:

- her address (`source` 1)
- the asset she will give (`give_asset1`)
- the quantity of `give_asset1` she will give (`give_quantity1`)
- the asset she will get (`get_asset`)
- the quantity of `get_asset1` she will get (`get_quantity`)
- how long before her order expires (`expiration1`)

Protocol-based trustless escrow

English

The Counterparty protocol acts as an escrow service, and thereby eliminates counterparty risk from the exchange of assets. Once Sally publishes her order `give_quantity1` of `give_asset1` is debited from her address; her address is debited *before* her order is matched with Alice's, and so she cannot spend those funds before `expiration1` passes, i.e. until her order expires. In the meantime, Sally's funds are not lost or borrowed, they are held by the protocol itself. If another order is placed which satisfies Sally's order, the protocol matches them, and sends each counterparty its respective funds.

Automatic order matching on the Bitcoin blockchain

`give_quantity1 / get_quantity1` is the "ratio" in which Sally will exchange `give_asset1` for `get_asset1`, and is denoted by `ratio1`. In order for two orders to be matched, `ratio1 must always be''greater than or equal'' to the inverse of` ratio2 `, Thus, if, for example` ratio2 (give_quantity1 + 1) / get_quantity1 `would be high enough ratio to match Sally's bet, but if` ratio2 =(quantity2 - 1) / quantity2 `it would not. Having been matched, the exchange is always made at` ratio1`. Further, when when an order is matched, the exchange is always settled as much as it can be.

A straightforward case

Suppose that Alice places order2 before `expiration` 1 which matches order1 perfectly: `give_quantity2 == get_quantity1` `get_quantity2 == give_quantity1`. Once Alice has made her order, the protocol debits `quantity2` of `asset2` from her address, and, since her order satisfies Sally's, Alice's order funds are sent to Alice, and Sally's order funds are sent to Alice. This completes the trade between Alice and Sally.

Matching an order: partially fulfilling an order

For the following example, let `give_quantity1 = 10` and `get_quantity1 = 20`, and that neither `give_asset1` nor `get_asset1` is BTC. Suppose that Alice wants to match Sally's order, does not want all 10 of `give_asset`; rather, she only wants 8.

Since the `ratio1 == 10/20 == 1/2`, Alice must `ratio2 >= 2/1`, to match Sally's order. In other words Alice must offer ''at least''16 of `asset_2` to get 8 of `asset_1` from Sally's order. Let's say Alice constructs order2 such that `give_quantity2 == 18` and hence `ratio2 = 18/8 > 2/1`.

English be settled at `ratio1`: for every unit of `give_asset1` that Sally gives Alice, she will get two units of `get_asset1` . Moreover, since every trade is settled as much and `give_quantity2 == 18` Sally will receive"18" `get_asset1` in exchange for 9 `give_asset1` .

Trading BTC on the decentralized exchange

Suppose Sally makes an order to trade `asset` in exchange for BTC, and Alice makes an order to trade BTC in exchange for `asset` . Upon placing order1, Sally's account is immediately debited, as usual, and, once Alice has placed `order2` , it is matched with `order1` . However, her BTC is not debited from her account, and the protocol will not send her Sally's XCP until Alice sends her BTC using Counterparty's `btcpay` function. If Alice sends the BTC using `btcpay` in "fewer than 10 blocks", the protocol will send her the XCP and thereby complete the transaction, otherwise, the trade expires, and the protocol will re-credit Sally's address with `give_asset` . This feature is enabled on the CLI, and disabled on Counterwallet, due to incompatibility with the browser-based security model.

Use-cases

Below are just a few of the many uses of assets, feel free to propose new uses if you find them.

Programmable Smart Contracts

Turing-complete smart contracts scripting is one of the most powerful Counterparty features. Users can write their own custom financial instruments and decentralized applications (Dapp). Counterparty contracts are 100% compatible with Ethereum scripting, and pretty much all contracts can be run on both platforms without code changes.

Currency Peg

Betting

Counterparty turns the Bitcoin blockchain into a betting platform and prediction market. Oracles can create broadcasts of information, and users can then place bets on these broadcasts. Funds are escrowed automatically by the protocol, and benefit from being stored securely inside the Bitcoin blockchain. Funds placed on bets are be provably inaccessible until the bet is resolved or expires. Oracles can set a fee fraction to receive for their betting feeds, providing incentive to run their broadcasts.

Tickets & Coupons

Assets can be used as tickets to a music event, parking tickets, coupons, etc.

Token Controlled Access (TCA)

Token Controlled Access is the idea of granting access to private forums, chatrooms, games, projects or other social media based on the ownership of tokens. Different types of tokens represent different types of membership, and holders of that token can register and/or view the restricted content. To invite new users, smaller fractions of these tokens can be transfered. If the token is indivisible and scarce, it will limit the amount of users others are able to invite. These tokens are also publicly tradable on the DEX and therefore can have a monetary value, and/or one proportional to other types of these tokens.

Proof of Publication

Using broadcasts, users can publish timestamped information onto the Bitcoin blockchain. This makes it possible to verifiy that something has been posted at a certain time, and it cannot be deleted.

Crowdfunding

Counterparty assets can be used for crowdfunding. You can issue a certain amount of assets and sell these to start your project. Due to the high amount of trust involved, it is better to use a Counterparty-based crowdfunding platform which can perform due-diligence on your project. This will provide your users trust, and demonstrate the legitimacy of your project. There is nothing stopping you from doing this on your own, but users may rightfully be suspicious about your project.

Derivatives

You can back Counterparty assets with tangible goods, such as gold.

In-game Currency

To integrate your multiplayer game into the global economy, Counterparty assets can also be used as in-game currency.

Altcoin Migration

If you have an altcoin that seeks to fulfill a specific purpose, but do not wish to continue mining, you can migrate it to Counterparty with proof-of-burn.

Verifiable Voting

Counterparty supports voting through the use of user-created tokens. This means that you can post the terms and options of your vote as a broadcast, and let users vote on its outcome with full transparency by using tokens.

If you create a token (`EXAMPLE`), you can create any other tokens (such as EXAMPLEVOTE) and pay distributions of EXAMPLEVOTE to all holders of `EXAMPLE` in one single action. Create a distribution payment and choose EXAMPLEVOTE as the currency to distribute. This way, all holders of `EXAMPLE` will receive EXAMPLEVOTE in the amount you specify.

Now all you need are as many different Bitcoin addresses as there are choices in your poll. For example: one Bitcoin address for yes, one for no. To cast their votes, holders of `EXAMPLE` can then send the EXAMPLEVOTE they have received to whichever address they agree with. The results of the poll will then be public and verifiable on the Bitcoin blockchain, and can be visualized in a block explorer.