You can support the author of this document by donating to this address:
bc1qg99fg656zflvfffvs7pgzje4rxe6sh8dw4p9nl

Using Sparrow Wallet

with Ordinals and Inscriptions

Table of Contents

Synopsis

Many users are new to Bitcoin and want to be able to receive Inscriptions and manage them prior to their full node being synced. Setting up a full Bitcoin node and Ordinal client with index will be covered in a different guide. This guide focuses on use of Sparrow Wallet on the desktop to manage transactions, be able to receive ordinals tied to inscriptions, and send them to others.

Sparrow Wallet is not “Ordinal aware”, which means without taking special precautions in preparing transactions that deal with ordinals, you can inadvertently lose them. But it does offer detailed analysis and control so that you can see and manage what is going on without having to build transactions on the command line. A very small wallet with just a few transactions has been established and will be referenced throughout this guide. This wallet is “on-chain” and addresses and transactions can be checked in block explorers.

You can support the author of this document by donating to this address:
bc1qg99fg656zflvfffvs7pgzje4rxe6sh8dw4p9nl

By the end of this guide you should have a better understanding of labeling your transactions, and performing a variety of send transactions to ensure that the ordinal tied to an inscription ends up where you want it to.

Server Connection Settings

Sparrow has three different connection options to be able to lookup address information for wallets. From the File menu, you can choose Preferences, and then Server on the window that appears to see your connection choices.

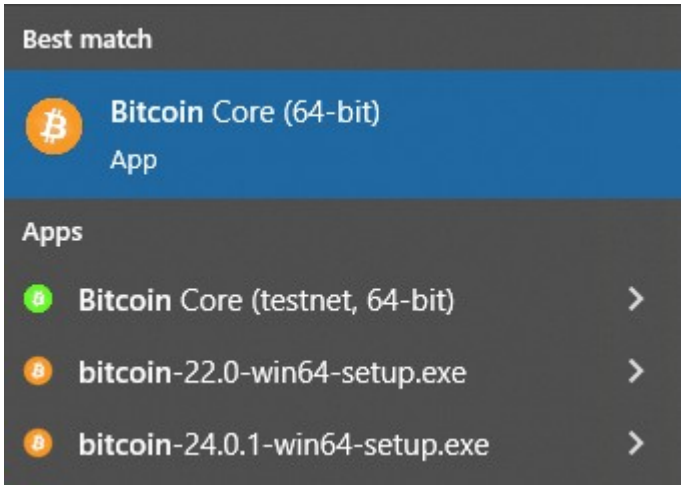
Many people start out with Public Server options but this means exposing information that you may not want to. For example, your IP address, the wallet addresses you lookup, can be correlated and kept indefinitely by the receiving service provider. And your internet service provider can infer that you're looking up Bitcoin related information from the server connected to.

If you've setup your own Bitcoin Core node or Private Electrum server, you should configure these settings now before proceeding. Personally I find the Private Electrum server, connected to an implementation running Fulcrum to be the most performant on my hardware. While outside of the scope of this guide, if you're interested in setting up a more robust Bitcoin node, consider Raspibolt and the guide for setting up Fulcrum on Linux here:
https://raspibolt.org/guide/bonus/bitcoin/fulcrum.html

If you are just starting out with Bitcoin, and have setup and running Bitcoin-QT, you should know that you may be able to connect to it from Sparrow by following these steps

1. Stop Bitcoin-Qt if it is running.

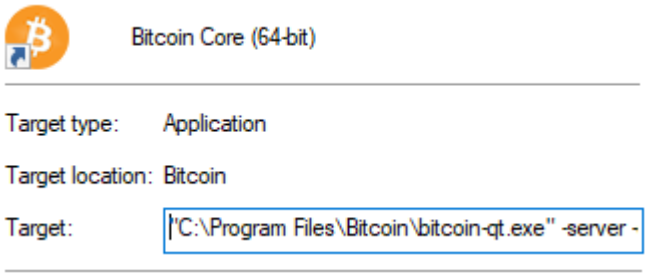
2. Find the shortcut to Bitcoin-QT. In windows, goto the Start menu, and begin typing Bitcoin



3. Right click on Bitcoin Core (64-bit) and choose Open File Location.

4. In the Windows Explorer dialog, right click the Bitcoin Core (64-bit) shortcut and choose properties

5. In the Target field, after the program, give a command line option for -server



6. Include any additional command line options. For example, my full Target value is as follows, because I wanted to store the Bitcoin Data on a different drive from my OS drive. This is all on one line.

```
"C:\Program Files\Bitcoin\bitcoin-qt.exe" -server -datadir=D:\Bitcoin1 -
rpccookiefile=D:\Bitcoin1\.cookie
```

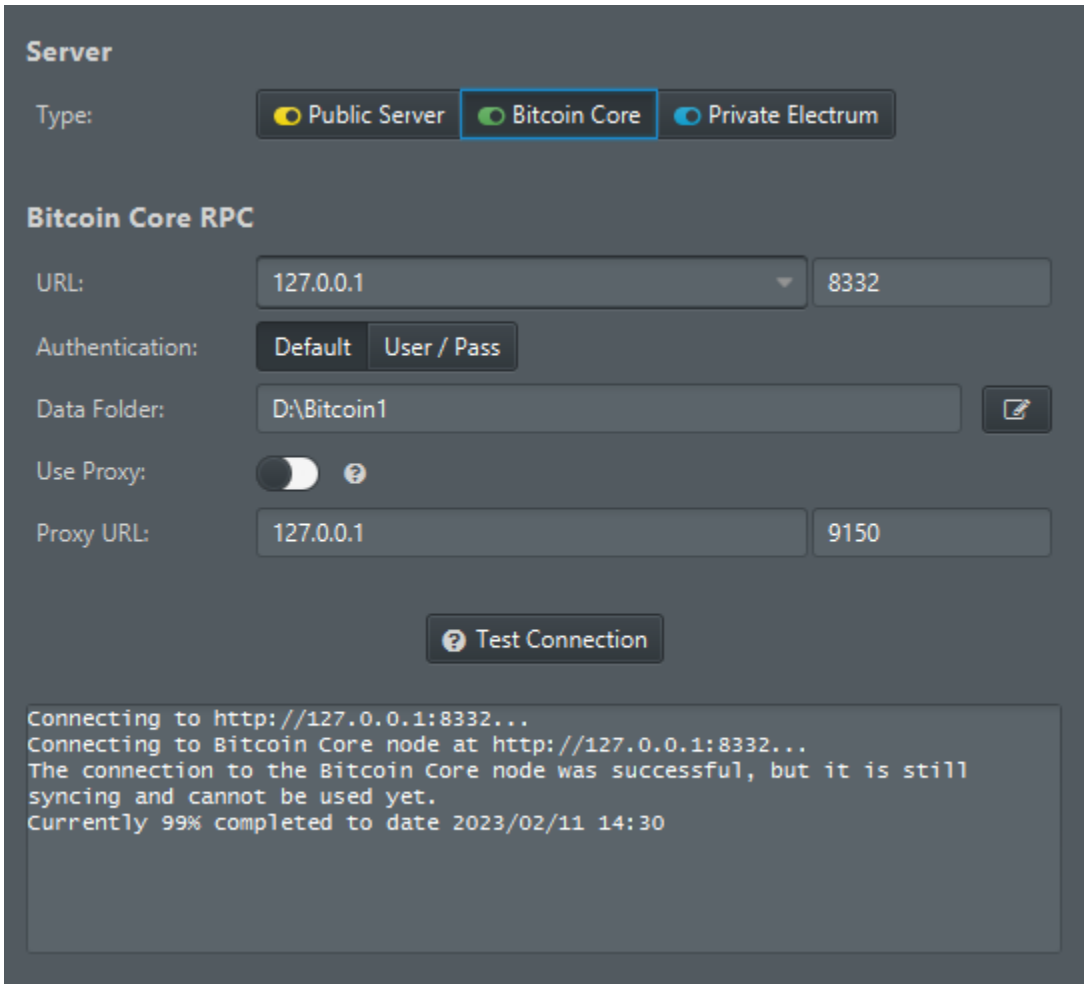
If you also store your Bitcoin Data in a different location than the default, then you may want to forcefully specify the location of the -rpccookiefile to reference that path as well. This is because Sparrow expects the .cookie file to be there, but Bitcoin-QT will default to creating in C:\Program Files\Bitcoin unless overridden. If Sparrow can't find the .cookie file, it will not be able to connect to Bitcoin-QT.

7. Press OK to close the shortcut properties window. Close Windows Explorer where the shortcut is.

8. Start Bitcoin-QT from the shortcut in the Start Menu by searching for Bitcoin again

9. Verify it has that option set. From the Settings menu, choose Options. You should see the "-server=" listed as an option overriding the dialog.

10. Back in Sparrow, choose the Bitcoin Core server type, and specify the connection settings as depicted. Only set the Data Folder if its different than the default.



11. You do not need to set the proxy for local calls, so leave that turned off.

12. Test the connection by clicking the button. If all is setup properly, it should give a result similar to what is depicted above. If you get a connection error, then recheck your settings in the shortcut used to launch Bitcoin-QT in step 6.

Importing seed words into Sparrow

While I would strongly encourage using a dedicated wallet for keeping track of your ordinals and inscriptions, some people want to import existing ones. A good use case of this is if you have started to setup Bitcoin and the Ord client, and created a wallet with Ord prior to indexing. When Ord creates a wallet, it provides a set of 12 seed words that can be used in the future to recover that same wallet.

As an example, assume you have the following 12 seed words.

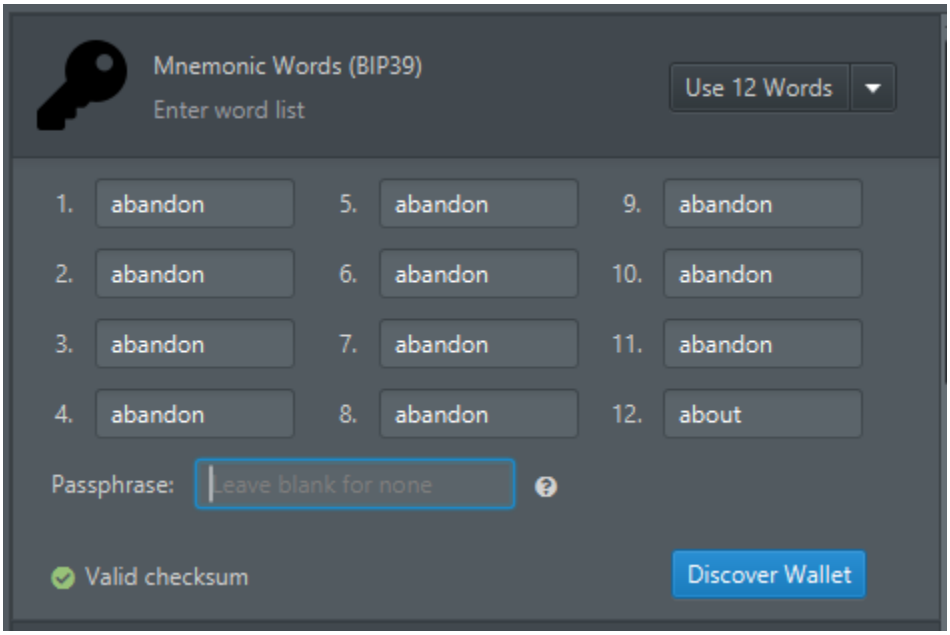
```
abandon abandon abandon abandon abandon abandon
abandon abandon abandon abandon abandon about
```

Note: This is a well known wallet. Do not use this for your own funds as multiple bots will sweep it out quickly. I like to refer to this wallet as the "socialist wallet". It's a useful wallet for this exercise to show transactions over time.

You can follow along recreating the wallet with Sparrow using these steps below before working with your own words to get more comfortable.

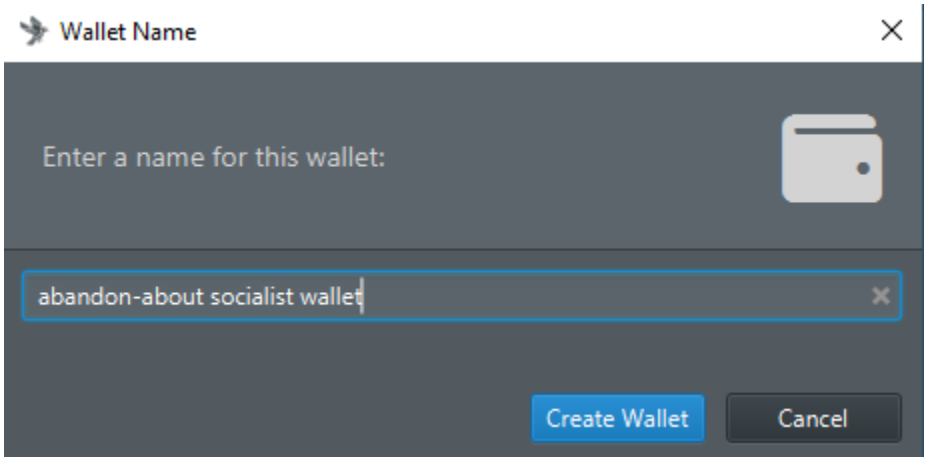
From the File menu, choose Import Wallet. A list of different import types will be presented. At the top of the list you'll see Mnemonic Words (BIP39). Change the selector from Use 24 Words to Use 12 Words. Enter the seed words in order. While entering the words, you'll see a warning in the lower left corner indicating "Invalid checksum". This is normal as the phrase is incomplete.

This wallet has no passphrase, so leave that field blank. A wallet created with Ord will also have no passphrase. If you are importing an existing wallet from another source, specify the proper number of words and provide the passphrase if the wallet has one. A passphrase here acts as additional entropy. Combined with the seed words, each passphrase produces a different wallet with different addresses. Note that after the final word has been entered, the warning has now changed to a green check with the words "Valid checksum".



Click the Discover Wallet. A few seconds will pass as assorted derived addresses for various address types for the wallet are checked to see if they have been used. If you import a wallet that has never seen a balance, you'll get a warning asking if you want to proceed anyway.

You'll then be prompted to save the wallet. Since this is a well known wallet used by others, I'll name it the "abandon-about socialist wallet".

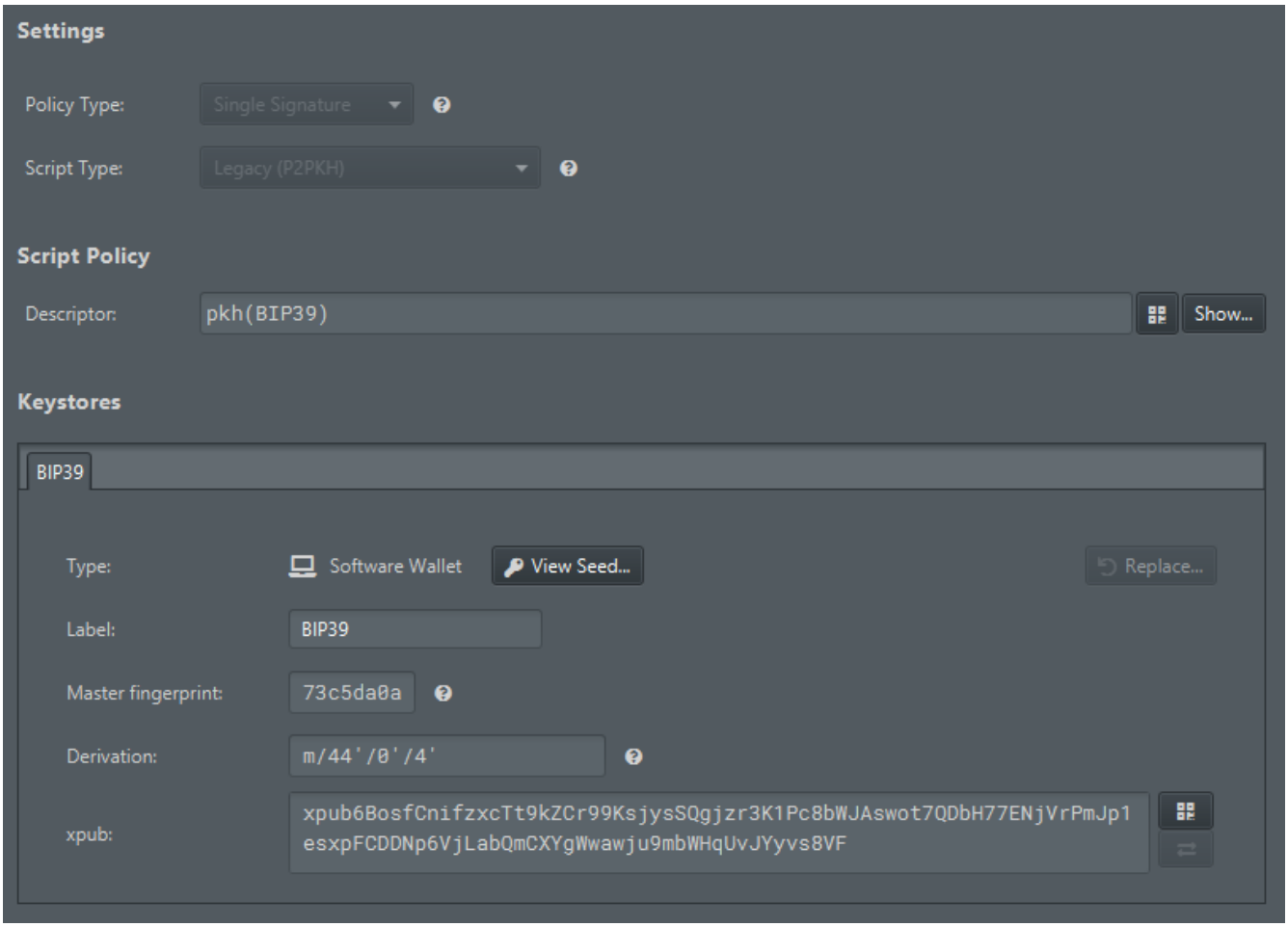


A subsequent screen will prompt for whether you want to add a password to the wallet. This is optional. It's important to note that this is NOT the same as a wallet passphrase. This password is used for encrypting the wallet file that Sparrow uses, and doesn't influence the master private key.

After completing those steps, you'll see the wallet transactions. By default, wallets use account 0. This wallet also makes use of accounts 1 and 4. A single master private key at the root of the wallet can be used for a myriad of accounts, each with their own addresses.

For this wallet, choose Account #4 on the left, and then click Settings at the bottom.

You can support the author of this document by donating to this address:
bc1qg99fg656zflvfffvs7pgzje4rxe6sh8dw4p9nl



This wallet is a legacy single signature pay to public key hash (P2PKH). The derivation path of m/44’/0’/4’ conveys this and is interpreted as follows:

- m – From the master node, derived from the master private key which is derived from the seed words
- /44’ – The purpose. Here, 44 is the standard derivation path for Pay to Public Key Hash (P2PKH) as defined in BIP 44.
- /0’ – The coin type. In this case, Bitcoin
- /4’ – The account

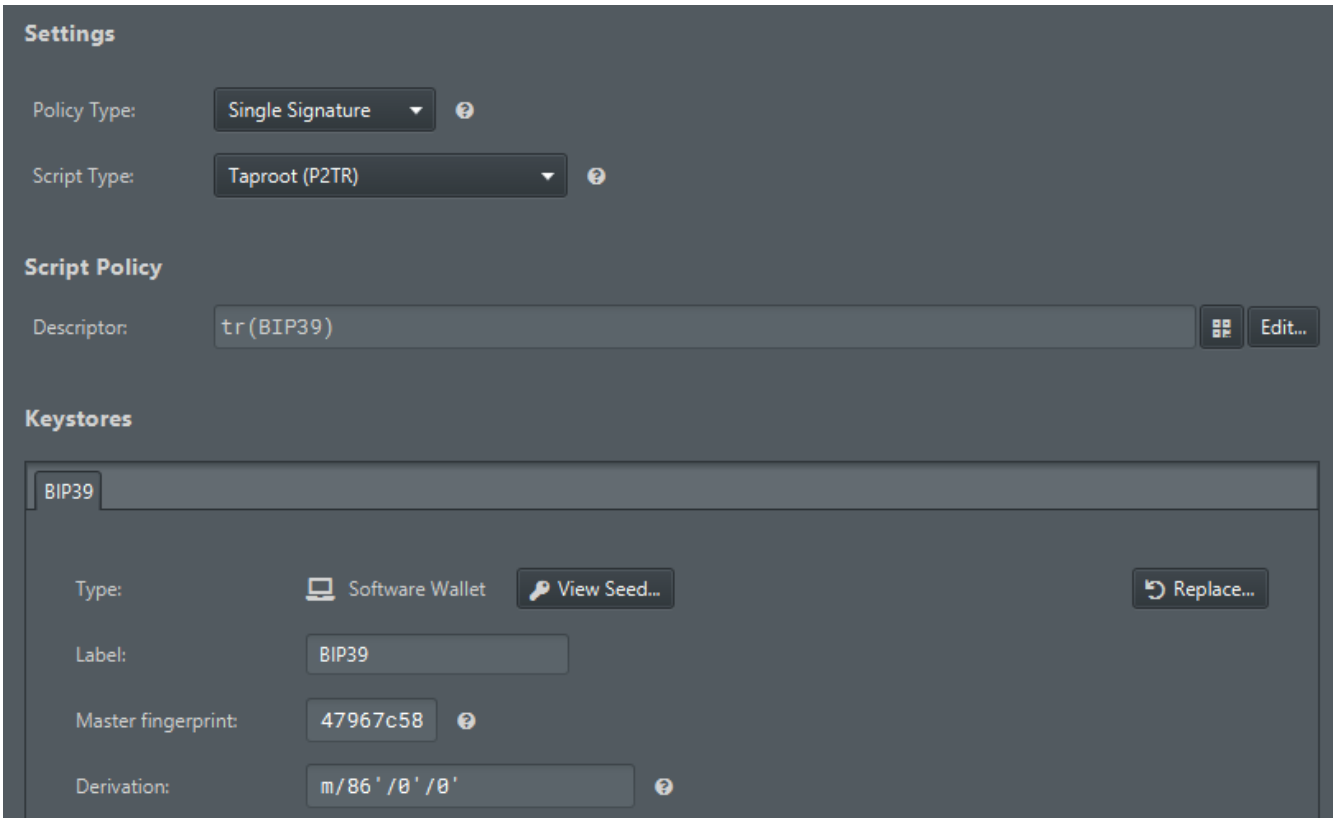
The xpub value for each account in a wallet is unique, and can be used to derive the addresses of the wallet.

Compare this to the settings view for the other accounts in this wallet.

You can support the author of this document by donating to this address:
bc1qg99fg656zflvfffvs7pgzje4rxe6sh8dw4p9nl

When you do an import of seed words where you are using other address types, the derivation may be different. For example, a wallet created with ord is focused on Taproot, and its derivation will begin with m/86', based on BIP 86, the definition for Key Derivation for Single Key P2TR Outputs.

Here is an example for a wallet that was first created with Ord, and then the seed words imported into Sparrow. See how the policy type and script type also differs for this wallet.



Click around the sections of the wallet to get more familiar. Proceed to the next section for an overview of labeling.

Labeling

It’s very important to keep track of your transactions over time. A wallet can get rather unwieldy quite quickly if you don’t record information and need it later. Labeling can help you manage privacy if you properly track where a transaction originated from, and sent to. Don’t assume that you’ll be able to remember what something was for, or who it came from. We all forget. Label your transactions, addresses, and UTXOs as you do them, and you’ll be better prepared.

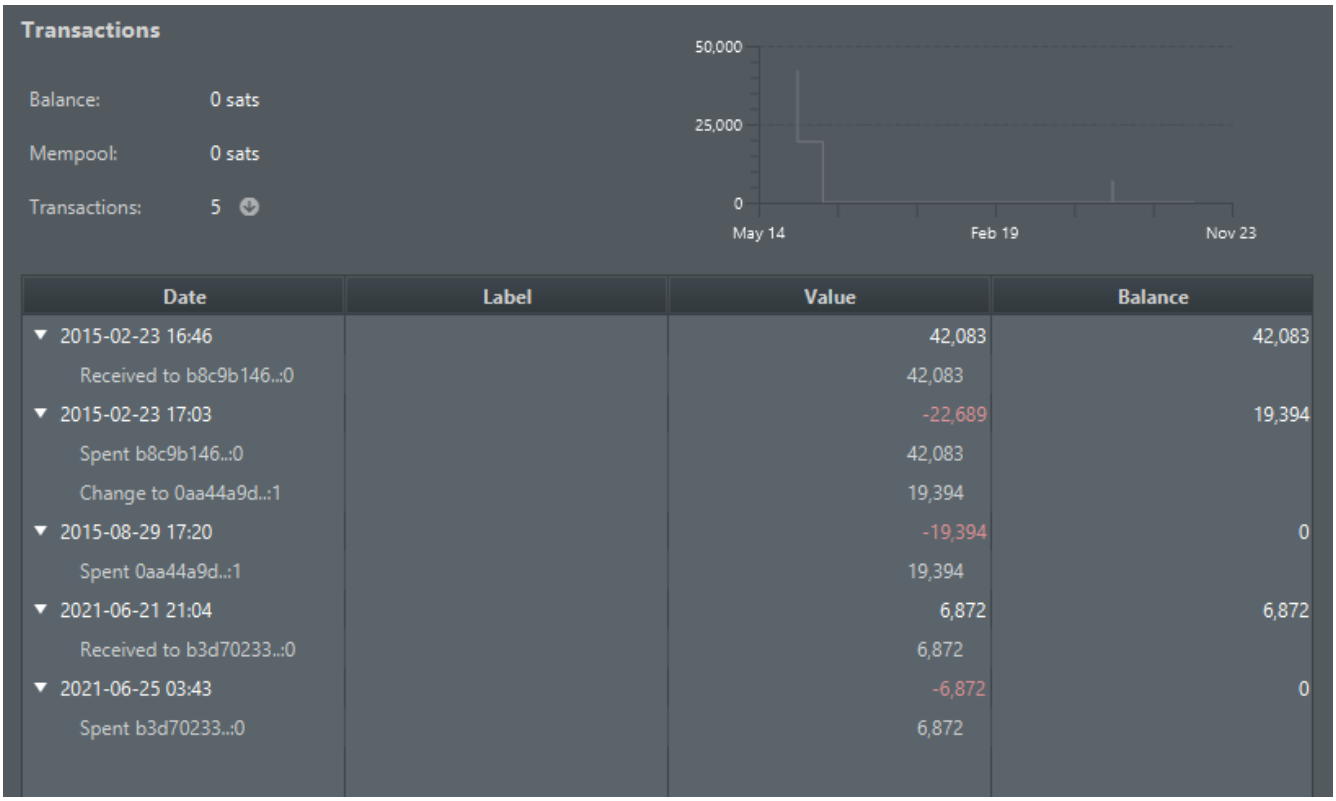
This section will use examples from Account #4 in the abandon-about socialist wallet.

Transactions

The transactions view provides the overall remaining balance of Bitcoin, the number of transactions, and by default lists the transactions ordered by date. This allows following chronologically through receives and spends for how it affects the balance.

The abandon-about socialist wallet shows transactions as follows. For clarity, I’ve expanded all the transactions by clicking the triangle to the left of each date, or double clicking the row. Note that there are no labels. Sparrow will provide a generic summary of the action denoting a Receive, Spent, and Change for each.

You can support the author of this document by donating to this address:
bc1qg99fg656zflvfffvs7pgzje4rxe6sh8dw4p9nl



We can add labels to each of these transactions by clicking once on the cell for the label, and then again for editing. A double click may inadvertently collapse or expand the detail rows. It may be easier to first click the cell you want to edit, and then use the keyboard by pressing enter. Note that as you add labels, Sparrow will automatically assign some of the labels to other related transactions. For example, here I add a label “Donation from Bob”, and Sparrow did the following

- The prefix of the label was used for the detail row indicating a receive because it wasn’t labeled yet

- The transaction for 2021-06-21 was also labeled the same because it is on the same receiving address.

You can support the author of this document by donating to this address:
bc1qg99fg656zflvfffvs7pgzje4rxe6sh8dw4p9nl

Date	Label	Value	Balance
▼ 2015-02-23 16:46	Donation from Bob	42,083	42,083
Received to b8c9b146...:0	Donation from Bob (received)	42,083	
▼ 2015-02-23 17:03		-22,689	19,394
Spent b8c9b146...:0		42,083	
Change to 0aa44a9d...:1		19,394	
▼ 2015-08-29 17:20		-19,394	0
Spent 0aa44a9d...:1		19,394	
▼ 2021-06-21 21:04	Donation from Bob	6,872	6,872
Received to b3d70233...:0	Donation from Bob (received)	6,872	
▼ 2021-06-25 03:43		-6,872	0
Spent b3d70233...:0		6,872	

Because the address was reused, it could be from the same entity, or for the same purpose. So this labeling makes some initial sense. But what if that second donation was received from Jane? We can update the labels to provide this level of detail.

Date	Label	Value	Balance
▼ 2015-02-23 16:46	Donation from Bob	42,083	42,083
Received to b8c9b146...:0	Donation from Bob (received)	42,083	
▼ 2015-02-23 17:03		-22,689	19,394
Spent b8c9b146...:0		42,083	
Change to 0aa44a9d...:1		19,394	
▼ 2015-08-29 17:20		-19,394	0
Spent 0aa44a9d...:1		19,394	
▼ 2021-06-21 21:04	Donation from Jane	6,872	6,872
Received to b3d70233...:0	Donation from Jane (received)	6,872	
▼ 2021-06-25 03:43		-6,872	0
Spent b3d70233...:0		6,872	

I've gone ahead and added labels for all of these transactions with some made up activities.

You can support the author of this document by donating to this address:
bc1qg99fg656zflvfffvs7pgzje4rxe6sh8dw4p9nl

Date	Label	Value	Balance
▼ 2015-02-23 16:46	Donation from Bob	42,083	42,083
Received to b8c9b146...:0	Donation from Bob (received)	42,083	
▼ 2015-02-23 17:03	Bought a pencil	-22,689	19,394
Spent b8c9b146...:0	Bought a pencil (input)	42,083	
Change to 0aa44a9d...:1	Bought a pencil (change)	19,394	
▼ 2015-08-29 17:20	Paid for postage	-19,394	0
Spent 0aa44a9d...:1	Paid for postage (input)	19,394	
▼ 2021-06-21 21:04	Donation from Jane	6,872	6,872
Received to b3d70233...:0	Donation from Jane (received)	6,872	
▼ 2021-06-25 03:43	Paid for phone top up minutes	-6,872	0
Spent b3d70233...:0	Paid for phone top up minutes (in...	6,872	

One of the spends had multiple detail rows. The transaction on 2015-02-23 indicates funds were Spent, and Change was made. Sparrow keeps track of the amount of Bitcoin that is retained within the wallet as change whenever an input is used for spending that isn't all sent to an external address.

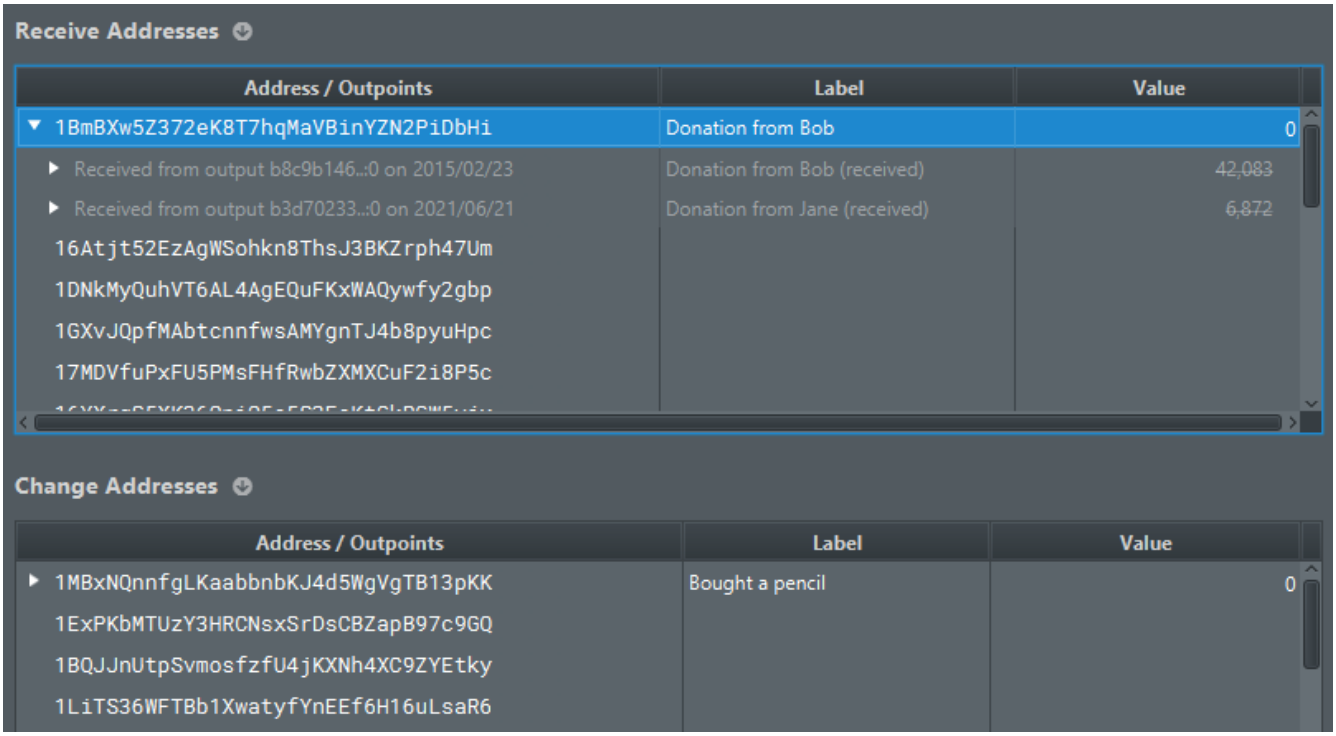
You can get as detailed as you want with your labeling. And perhaps your labeling style differs depending on intent of he wallet. For example, here is some labels of a wallet with ordinal inscriptions. In this case, an emphasis on the beginning part of the label has been made because it makes things easier when setting up transactions where coin control is important for ordinals. That will be covered in later sections.

Date ▲	Label	Value	Balance
▼ 2023-02-08 02:26	funding 1	20,750	20,750
Change to 5dc7cd0b...:0	received funds in wallet (change)	20,750	
▼ 2023-02-08 20:46	funding 2	239,338	260,088
Change to 6ccb34d2...:0	received funds in wallet (change)	239,338	
▼ 2023-02-09 15:27	test: inscribing #1 text file	-15,648	244,440
Received to dbf4e10...	change from inscribing #1	5,102	
Spent 5dc7cd0b...:0	commitment for inscribing #1	20,750	
▼ 2023-02-09 15:27	test: reveal #1 text file	10,000	254,440
Received to 517214...	38220 Inscription txt (inscription o...	10,000	
▼ 2023-02-09 22:20	test: inscribing #2 image file	-54,904	199,536
Received to 8a37865...	change from inscribing #2	184,434	
Spent 6ccb34d2...:0	commitment for inscribing #2	239,338	
▼ 2023-02-09 22:20	test: reveal #2 image file	10,000	209,536
Received to 93c75f1...	41492 Inscription jpg (inscription ...	10,000	

You can support the author of this document by donating to this address:
bc1qg99fg656zflvfffvs7pgzje4rxe6sh8dw4p9nl

Addresses

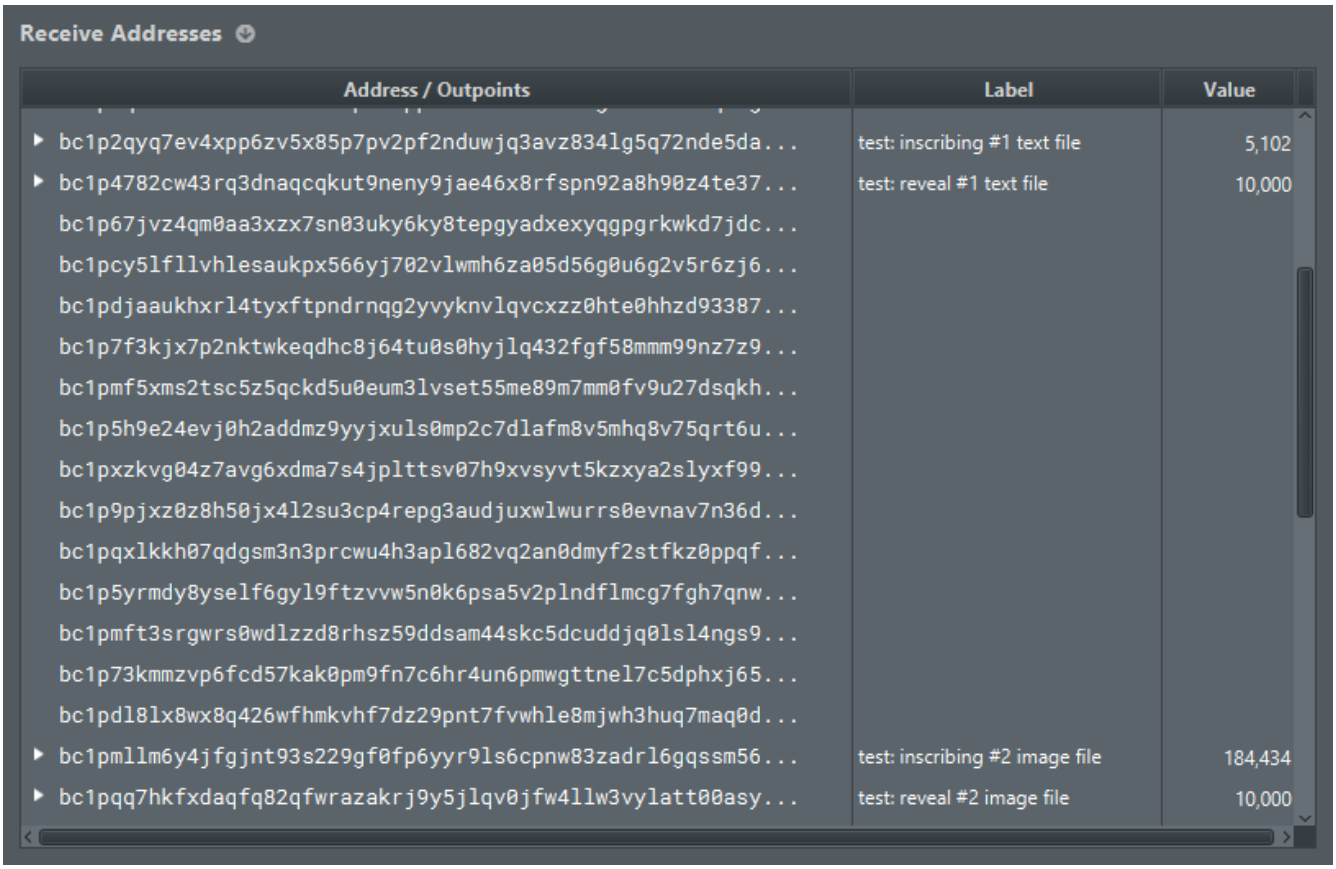
Once your transactions are labeled, move on to the Addresses view. This view focuses on just the addresses in your wallet, and will carry over the labels that were assigned during the transaction.



If you alter a label in this view, the address line will not be reflected anywhere else, but changes to the detail line will be. Rname the entry for “Bought a pencil” to “Bought a pack of pencils”. There will be no discernable change on the Transactions view. But if you relabel a detail record, which represents an output, then that will be changed in the Transactions view as well.

If you are looking at a wallet imported from Ord, you will notice that your addresses may have groupings, and possible gaps. See the following example of receive addresses:

You can support the author of this document by donating to this address:
bc1qg99fg656zflvfffvs7pgzje4rxe6sh8dw4p9nl



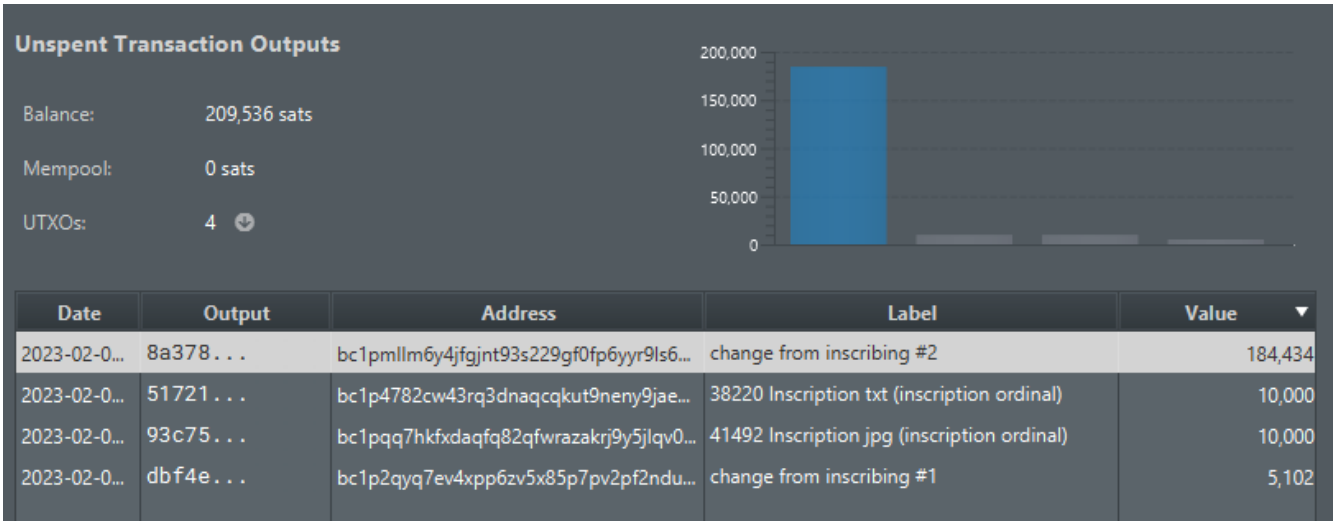
In this example, each inscription is followed by a reveal transaction. That's how the commitment and reveal transactions were labeled in this wallet on the transactions tab. The gaps present here are the result of using the `--dry-run` command line option for Ord multiple times between inscriptions. Each call advances address counters. Ord and Bitcoin do not reuse addresses by default.

UTXOs

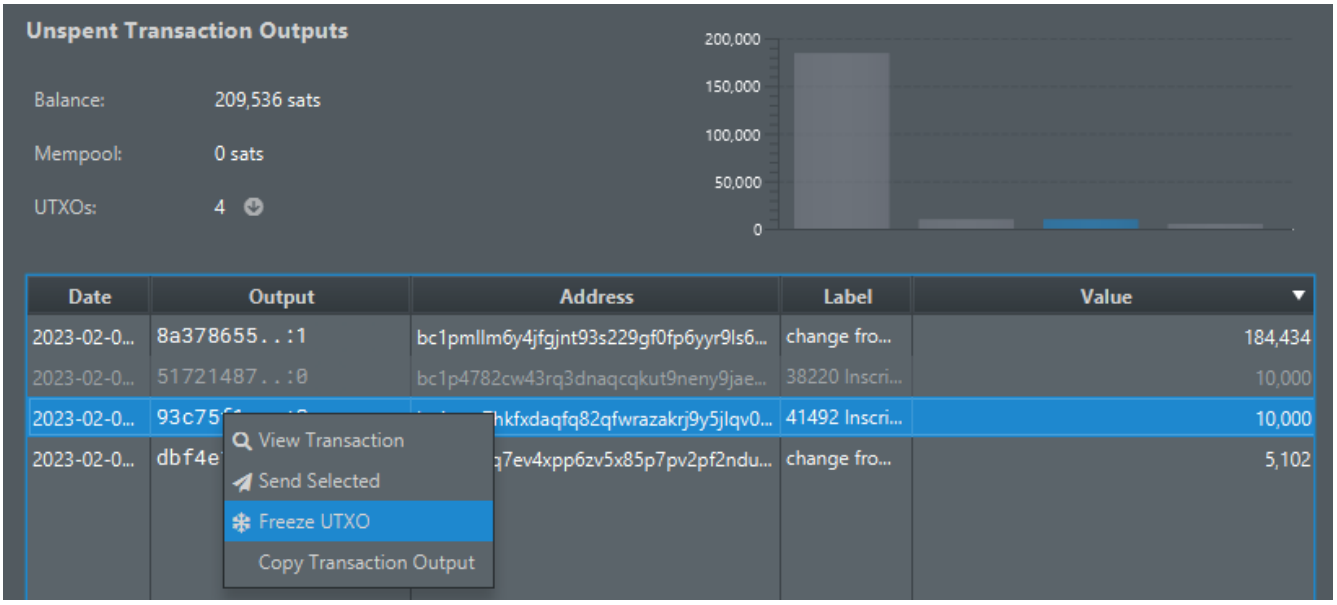
A UTXO is an Unspent Transaction Output. Sparrow Wallet allows for both labeling individual UTXOs and choosing them to spend from as a method of coin control. This will become quite important in later sections discussing how to spend from Sparrow, while ensuring Ordinal Inscriptions remain intact.

None of the accounts in the abandon-about socialist wallet has UTXOs at the time of this writing, so we'll take a look at the example Ord wallet. Here there are 4 current UTXOs. Two of them are from the inscriptions created, and two of them are resulting changing. This comes directly from the

information entered on the Transactions view. Making an alteration here will be reflected in the corresponding detail row of Addresses, and Transactions



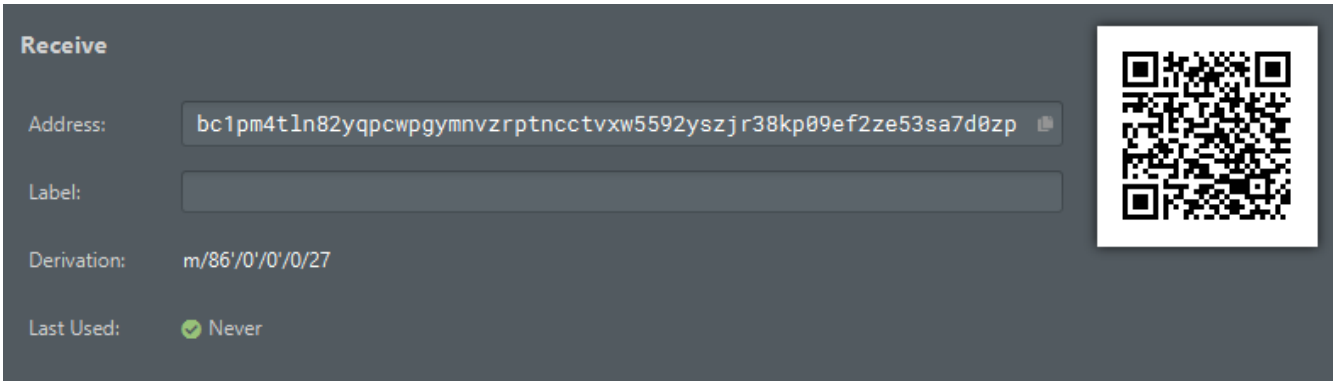
From the UTXOs view, you can right click an inscription output, and choose Freeze UTXO to avoid accidental spending. Frozen transactions will be greyed out.



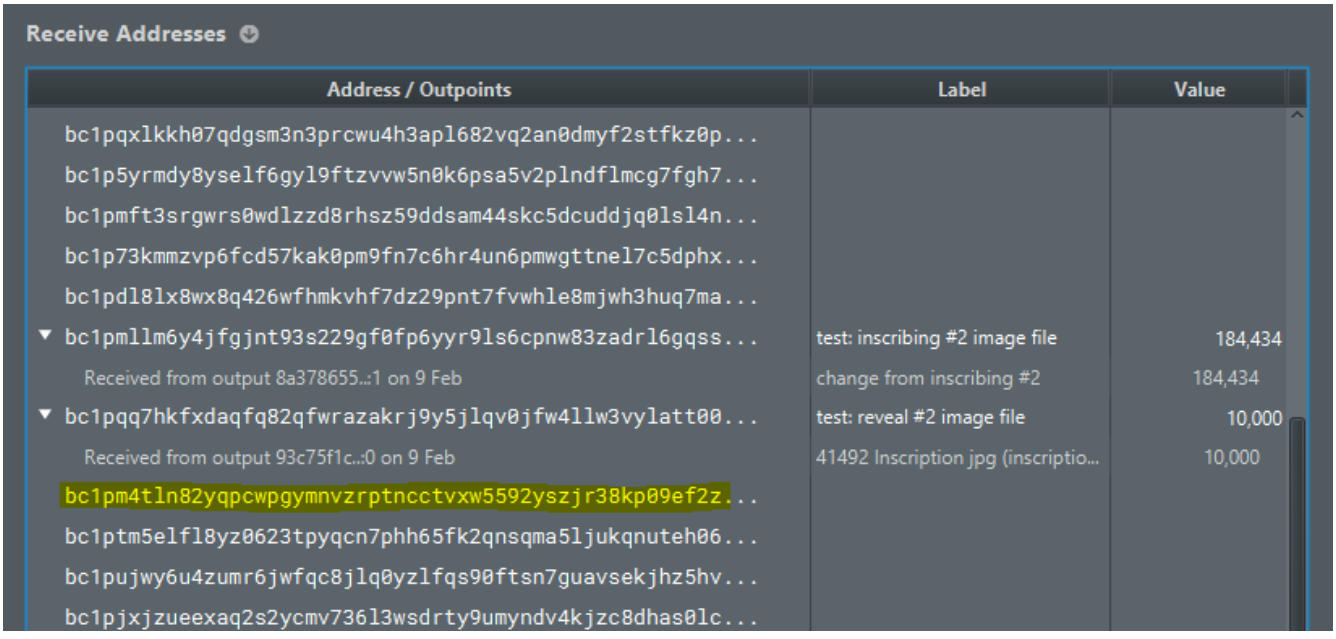
You can support the author of this document by donating to this address:
bc1qg99fg656zflvfffvs7pgzje4rxe6sh8dw4p9nl

Receiving Bitcoin and Inscriptions

To receive Bitcoin, whether it references an Inscription or not, you can view the Receive tab. By default, the next unused address will be shown, along with a corresponding QR code. You can label the transaction on this screen. Sparrow provides other useful details about the derivation path and descriptors, but you can ignore those for now. If someone asks for a Bitcoin address to send to, this is the quickest way to get to it.



You can cross check the list in Addresses view verify that the address referenced is the next one.



When receiving an Inscription, be certain to label the address so that you can properly reference it later.

You can support the author of this document by donating to this address:
bc1qg99fg656zflvfffvs7pgzje4rxe6sh8dw4p9nl

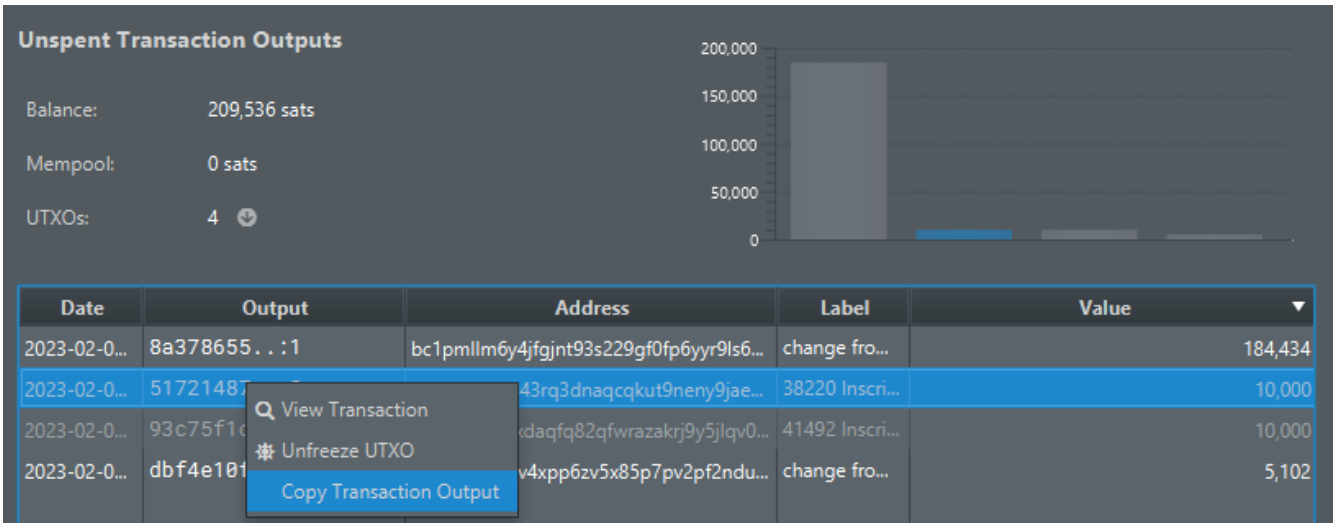
Remember. From the UTXOs view, you can right click an inscription output, and choose Freeze UTXO

Using Ordinal Explorer to View Inscriptions

While outside of the scope of Sparrow you should be aware of ways to view inscriptions and verify them based on what you have in your wallet. The ordinals website provides functionality to lookup an inscription at https://ordinals/inscriptions/

In the search bar, you can search for your inscriptions by transaction id or transaction output.

From Sparrow, you can get the transaction output by right clicking on a UTXO that is bound to an inscription, and choosing Copy Transaction Output.



This transaction has an output of
51721487e92ecc3a1a0ead8614e8cdee7a221b30e65af6aa2a58d3a097a1134d:0

Search for it at the top of the ordinals website, or navigate directly to it at
https://ordinals.com/output/51721487e92ecc3a1a0ead8614e8cdee7a221b30e65af6aa2a58d3a097a1134d:0

The inscription referenced can be found here
https://ordinals.com/inscription/51721487e92ecc3a1a0ead8614e8cdee7a221b30e65af6aa2a58d3a097a1134di0

Sending an Inscription to Another Address

To send an inscription from within Sparrow Wallet to another address, you must take special care or else you may accidentally lose the inscription by sending it to mining fees, the wrong address, or have

the ordinal output be found in the middle of an address. This section will cover assorted scenarios, but the general rules to follow are as follows

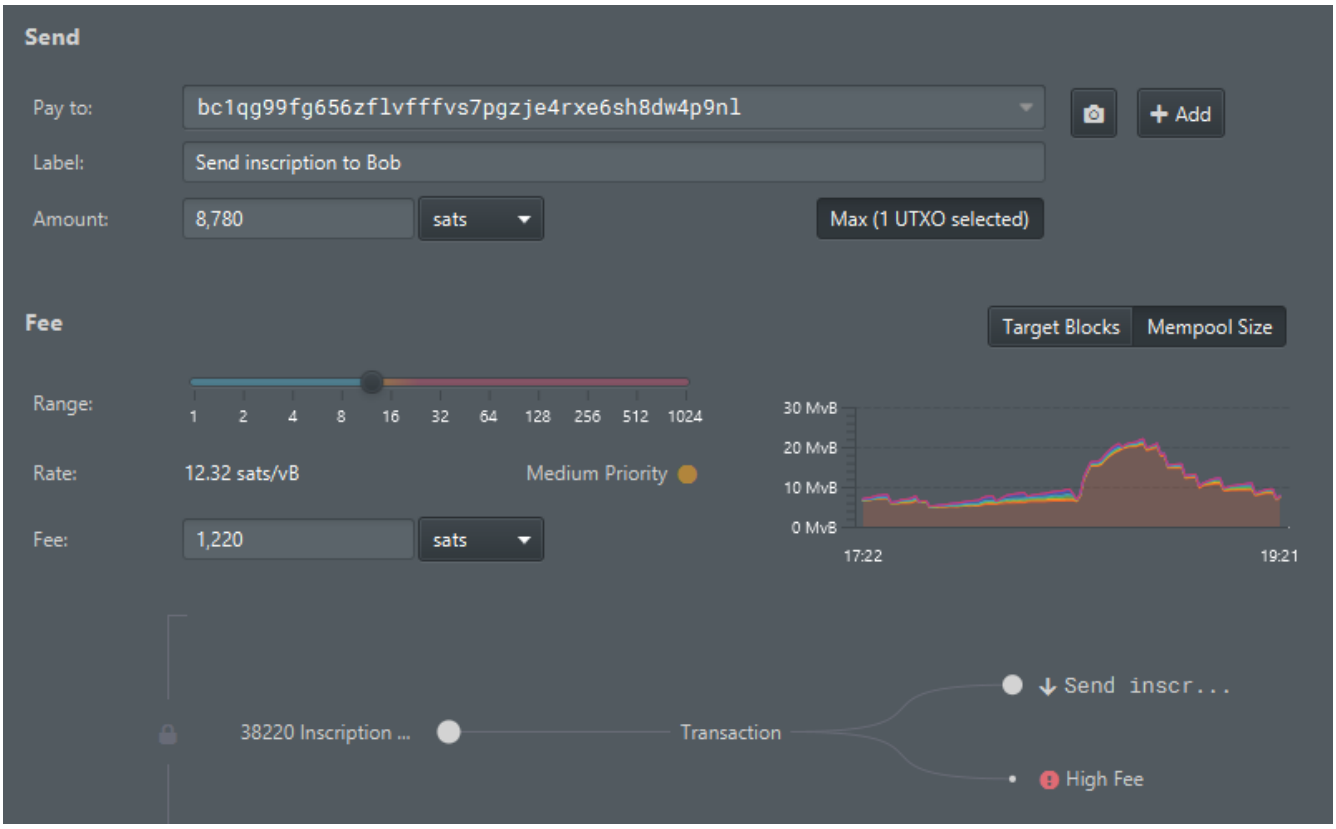
1. Only transfer one inscription at a time. While possible to send multiple inscriptions to multiple different addresses, that is outside of the scope of this guide.
2. Always have the inscription being sent listed as the first input
3. Always have the address where the inscription should be sent to as the first output
4. Verify the structure of the transaction inputs and outputs ordering at each stage of the process before broadcasting

1 Input – 1 Output

The simplest method of transferring an inscription to another address is if there is only one input, and one output. In this case, fees for transaction are paid from the remaining value on the inscription. Inscriptions that are created using Ord will start with 10000 sats output value for the specific purpose of facilitating transfers like this.

Within Sparrow, from the UTXOs tab, find the inscription you want to send. Select just that row. Then click the Send Selected button in the lower right corner. The send tab will be initialized. Provide the receiving address in the Pay to field. Provide a label. In this case, I’m sending the inscription to Bob.

You can support the author of this document by donating to this address:
bc1qg99fg656zflvfffvs7pgzje4rxe6sh8dw4p9nl



Click the Max (1 UTXO selected) button, and adjust fees accordingly. You should be able to easily verify that the total of the amount that will be sent + the fees will equate to the UTXOs value. The bottom part of the screen shows the structure of the transaction, 1 input on the left, 1 output on the right, and a “High fee”. Click the Create Transaction button to continue. Review the transaction overview which will be the same as you saw of the bottom of the previous screen. Click Finalize Transaction for Signing. Verify the same order of inputs and outputs are shown (for a 1 input to 1 output they will always be the same order). Sign the transaction. You can review again prior to broadcasting. Once it is sent, it’s much more challenging to correct any errors, and virtually impossible if you sent to an address outside of your wallet.

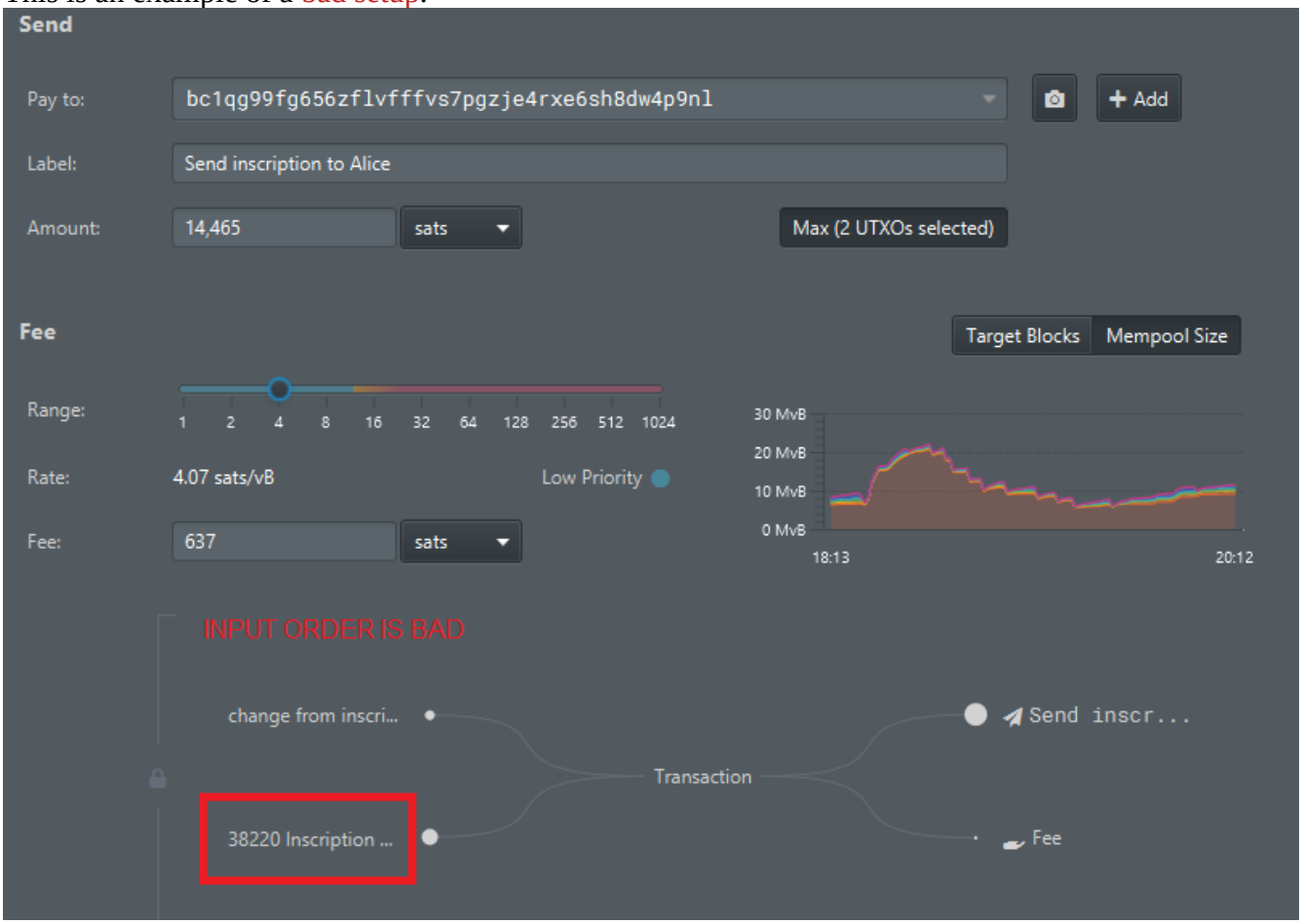
Many Inputs – 1 Output

This is a common scenario where you either want to add additional value to an inscription, or pay fees for the inscription from additional UTXOs so as not to reduce the value assigned the inscription.

Within Sparrow, from the UTXOs tab, find the inscription you want to send. Select just that row. For each additional UTXO you want to add to the transaction, idnentify where it is in the list, and control+click it. You should now have 2 or more selected UTXOs and can click the Send Selected button in the lower right corner. The send tab will be initialized. Provide the receiving address in the Pay to field. Provide a label. In this case, I’m sending the inscription to Alice. Note that as you adjust

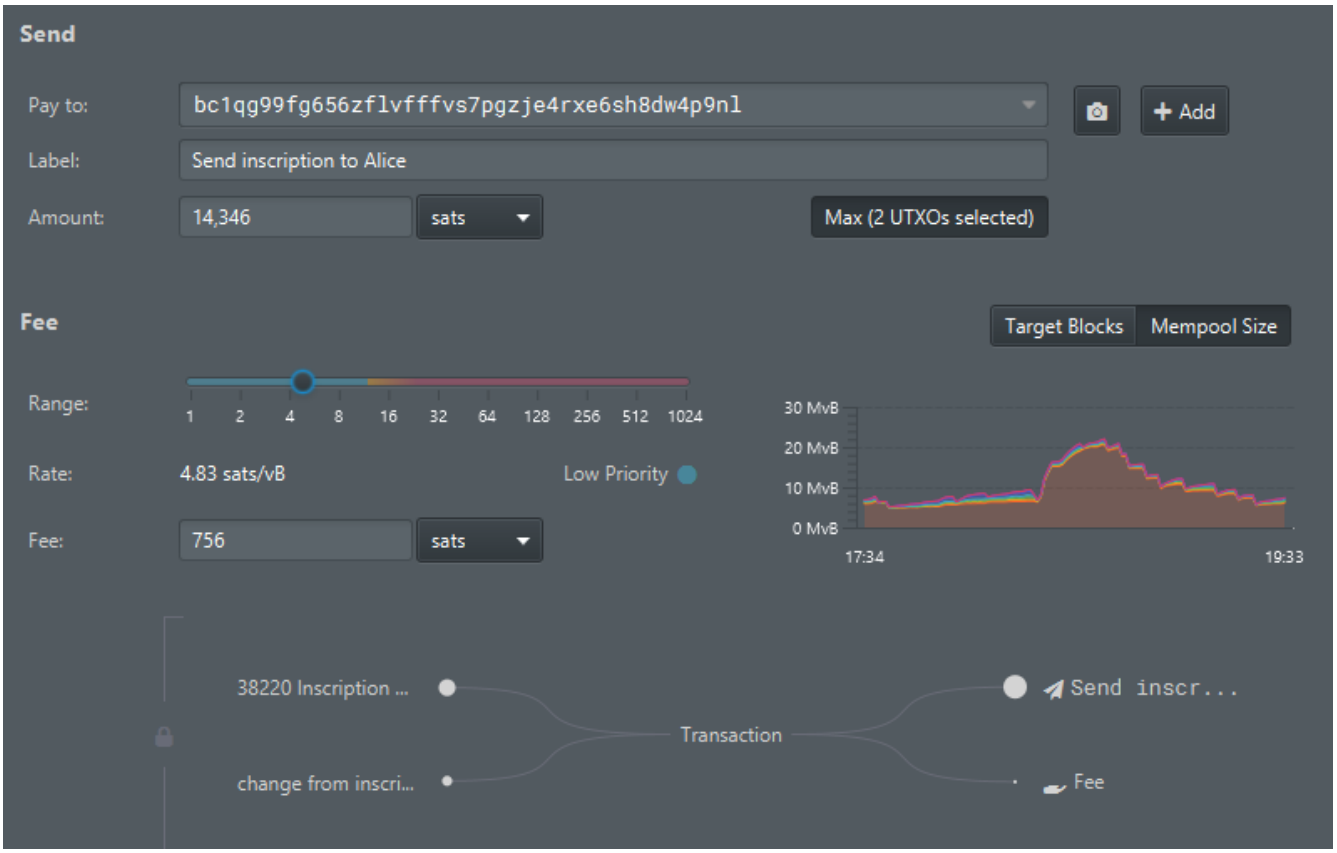
settings, Sparrow will rebuild the transaction and depict the order of inputs on the left side. It is very important that the inscription being sent be listed as the first input.

This is an example of a bad setup.



You can adjust the slider for fees paid to cause Sparrow to depict a new version of the transaction. Here is an example of the proper order, listing the inscription at the top

You can support the author of this document by donating to this address:
bc1qg99fg656zflvfffvs7pgzje4rxe6sh8dw4p9nl



With the inscription at the top, click Create Transaction button. Verify that the order of inputs is as desired. You can hover over the transaction outputs to see the amounts referenced. Remember, ordinal outputs are on outpoint 0, and when created with the Ord tool, inscriptions start with 10000 sats.

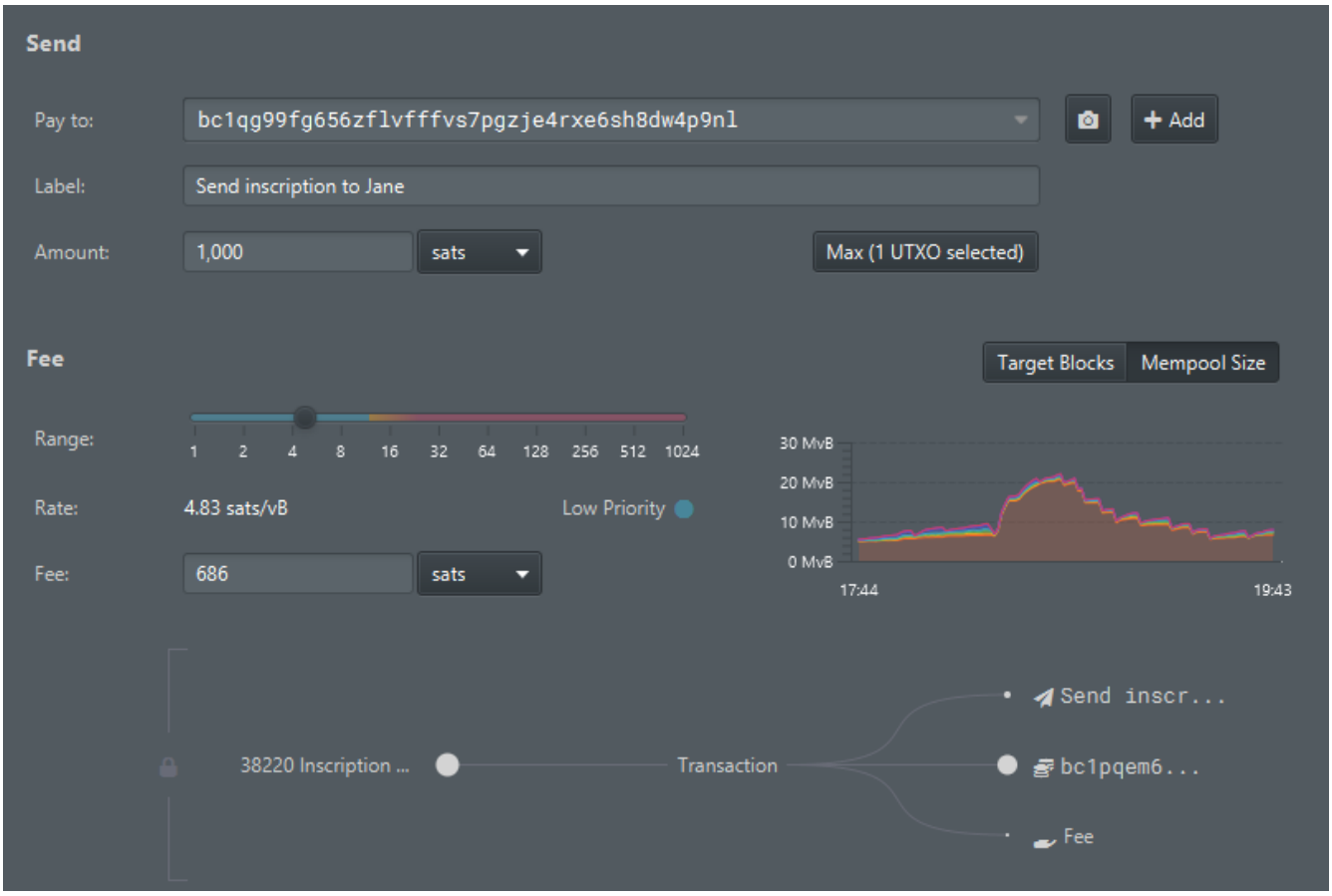
When satisfied, click the button to Finalize Transaction for Signing, Sign and Broadcast.

1 Input – Many Outputs

In this scenario you may want to retain value from an inscription instead of sending all of the sats. While less common then others, its important to know how this is done and to show controlling of the output side of the transaction.

Within Sparrow, from the UTXOs tab, find the inscription you want to send. Select just that row. Click the Send Selected button in the lower right corner. The send tab will be initialized. Provide the receiving address in the Pay to field. Provide a label. In this case, I'm sending the inscription to Jane. I reduce the amount down to 1000 sats. Sparrow will rebuild the transaction anytime I change the pay to address, amount or fees. Having reduced the amount to pay, Sparrow will build the transaction with a change address for the remainder.

You can support the author of this document by donating to this address:
bc1qg99fg656zflvfffvs7pgzje4rxe6sh8dw4p9nl



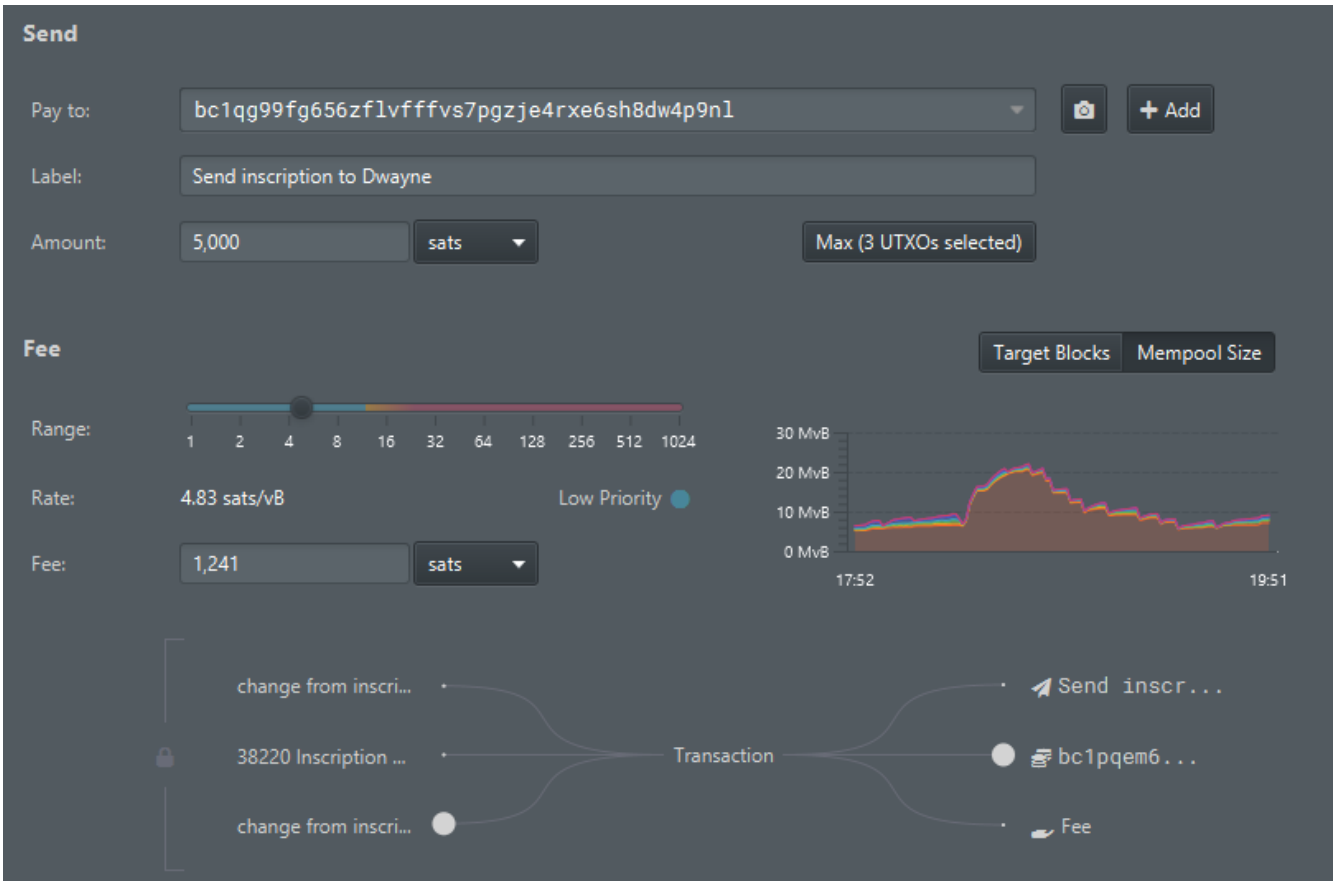
It is very important that the inscription being sent be listed as the first output on the right hand side. If it is not the first output, When satisfied, press the Create Transaction button. Ensure the order remains as expected, Finalize Transaction for Signing, Sign it and Broadcast.

Many Inputs – Many Outputs

One of the benefits of Bitcoin is doing multiple activities together as a batch to save on transaction fees. In this example, we'll send an inscription, include all of the change UTXOs, and include additional payment addresses outside of the wallet while consolidating.

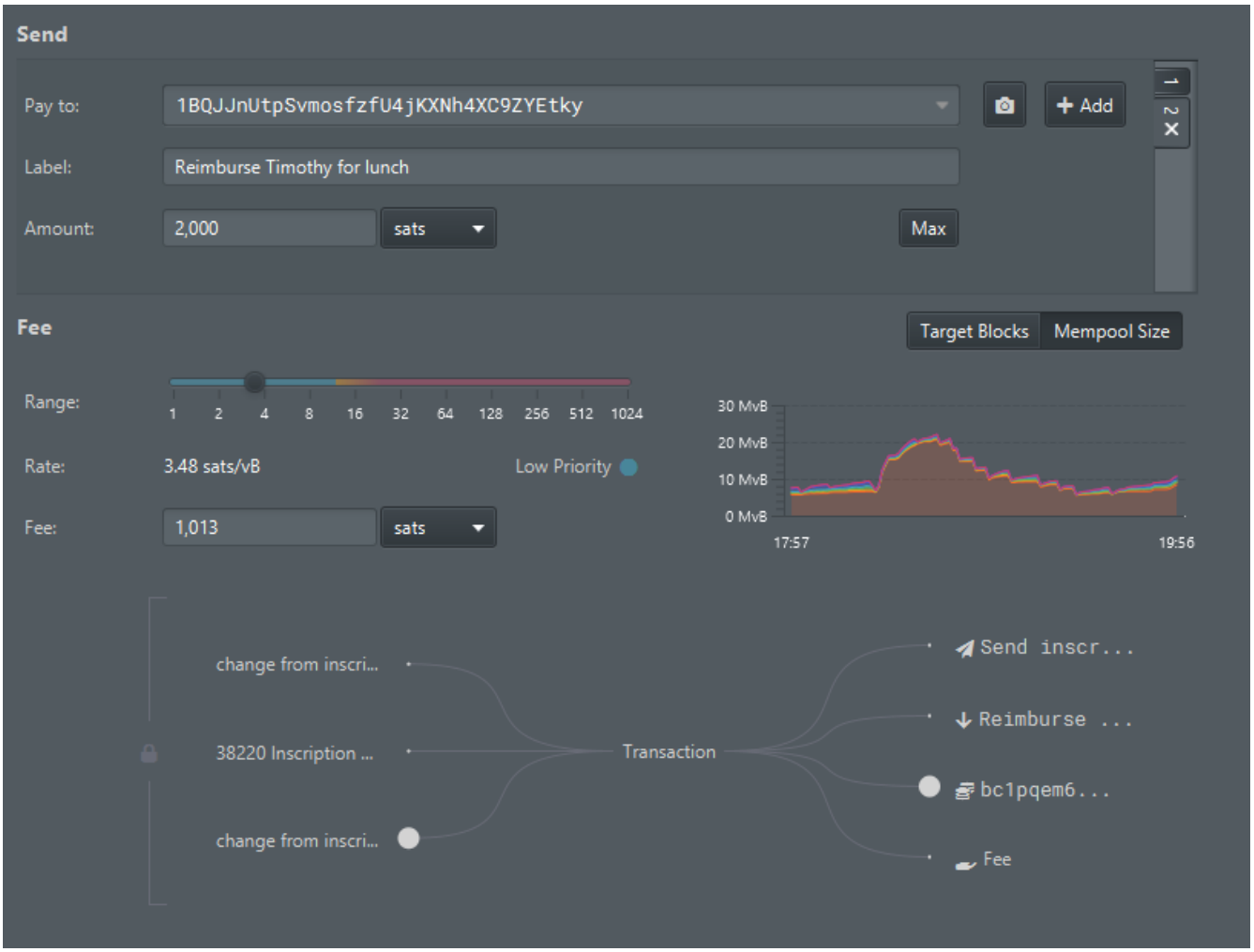
Within Sparrow, from the UTXOs tab, find the inscription you want to send. Select just that row. For each additional UTXO you want to add to the transaction, idnentify where it is in the list, and control+click it. You should now have 2 or more selected UTXOs and can click the Send Selected button in the lower right corner. The send tab will be initialized. Provide the receiving address for the inscription in the Pay to field. Provide a label. In this case, I'm sending the inscription to Dwayne, and setting the amount to 5000 sats.

You can support the author of this document by donating to this address:
bc1qg99fg656zflvfffvs7pgzje4rxe6sh8dw4p9nl

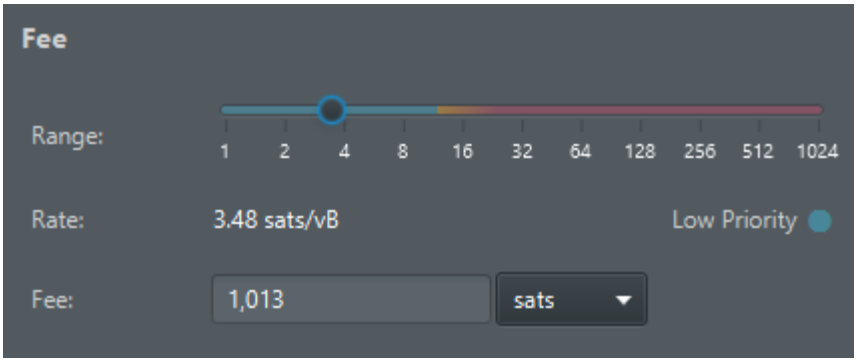


My initial setup looks like this. Already I know that the input side will not be suitable as the inscription isn't the first input, so I'll need to make adjustments. I also want to pay Timothy 2000 sats in this transaction. At the top of the transaction to the right of the address I'm using for Dwayne, I can click the Add button and provide Timothy's address and indicate the amount of 2000 sats. Sparrow has now suggested this transaction, which still has an unusable input order to maintain inscription control.

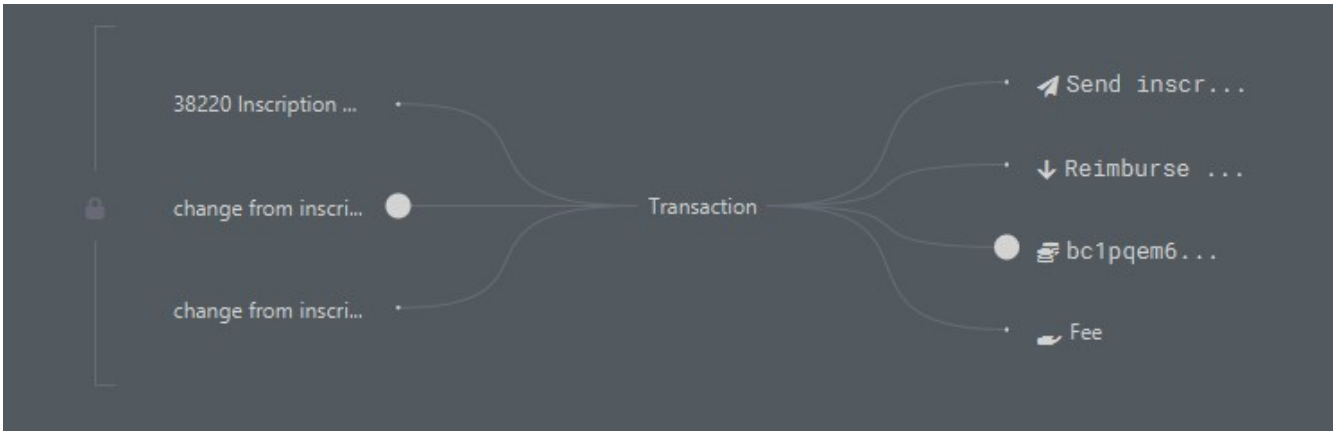
You can support the author of this document by donating to this address:
bc1qg99fg656zflvfffvs7pgzje4rxe6sh8dw4p9nl



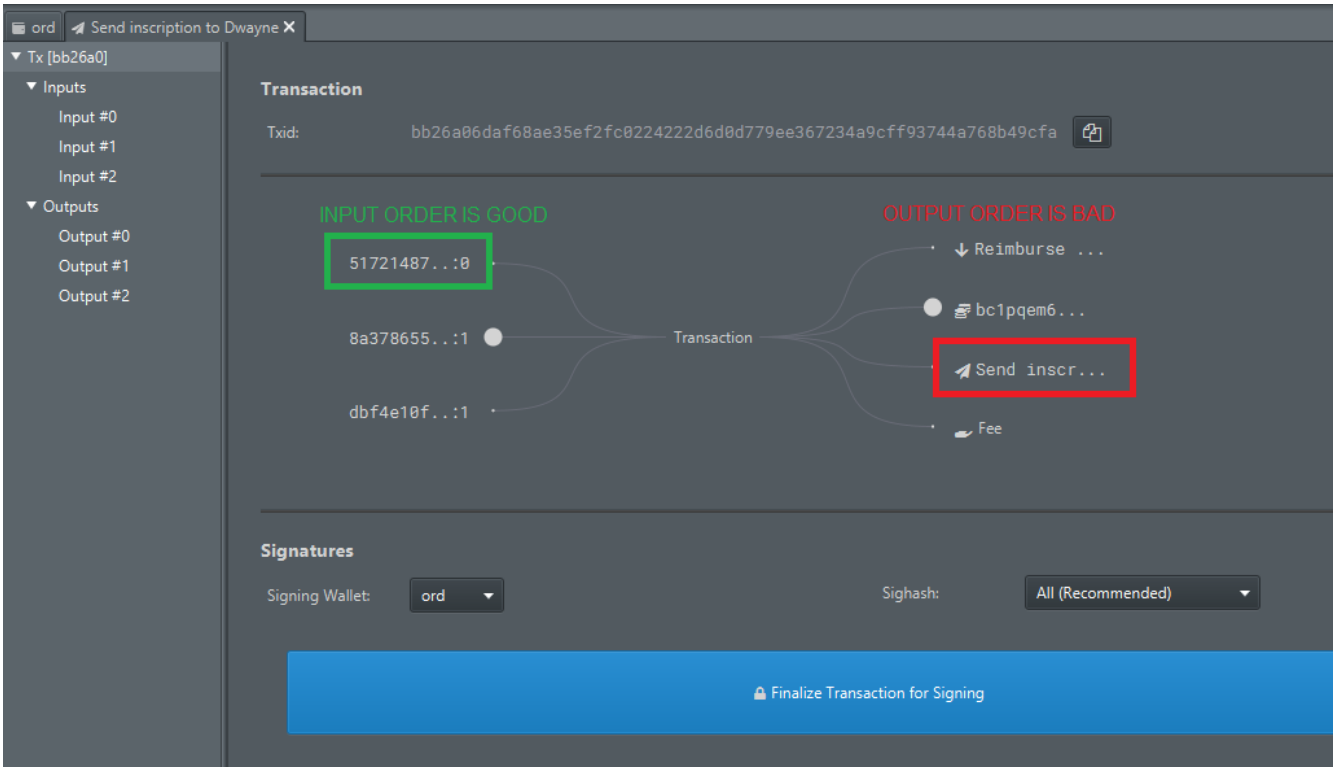
The easiest way to force Sparrow to rebuild the transaction without changing address information is to adjust the fee slider, as this will alter the amount being sent to our change address requiring the transaction to be recreated



After a few adjustments, I have my desired structure. On the input side the inscription is the one at the top. And on the output side, the inscription is being sent to the one on the top.

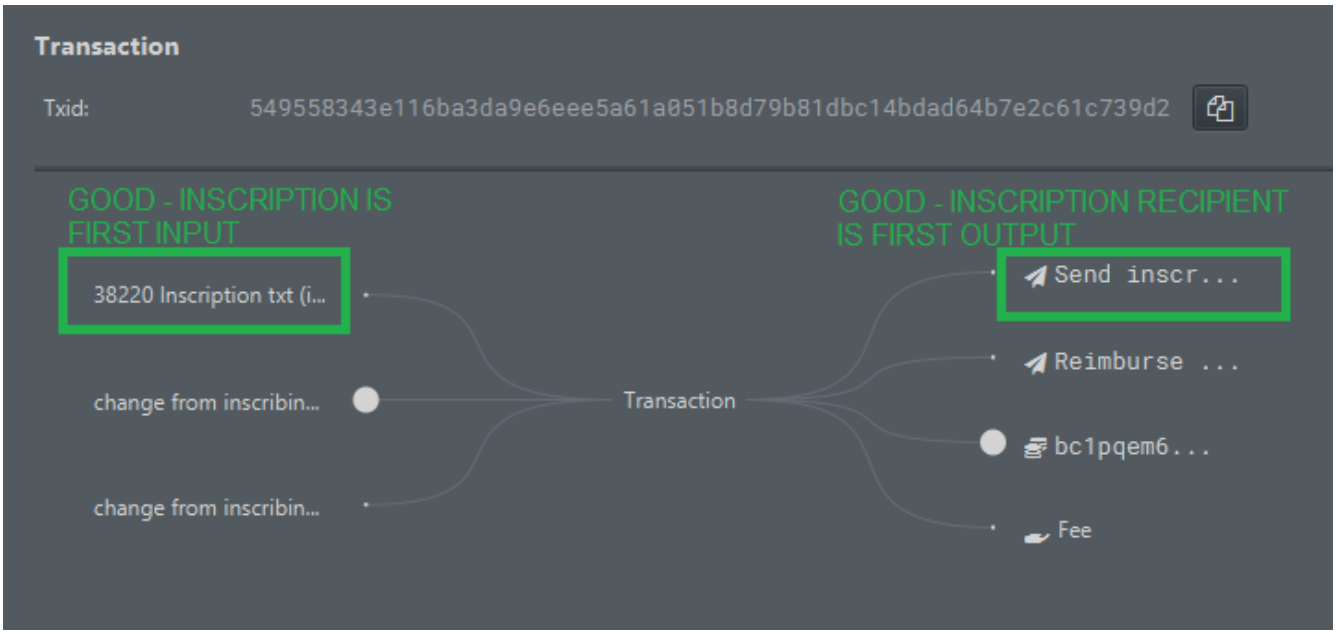


Click the Create Transaction button. Now unfortunately, the transaction gets recreated here, and the order can change. I can see that the newly made transaction is unsuitable. So close the tab for this transaction, and return to the Send view of the wallet.



You can support the author of this document by donating to this address:
bc1qg99fg656zflvfffvs7pgzje4rxe6sh8dw4p9nl

The Send view retains the settings as before, so you should be able to just click the Create transaction button again. Repeat the process until the view on the Transaction screen is ideal.



Once its in the right order, click Finalize Transaction for Signing. The order should not change, but visually verify it anyway. Sign the transaction and Broadcast.

Conclusion

As you can see, Sparrow Wallet, even without being Ordinal and Inscription aware, can still be used to safely manage your inscriptions. It just takes a bit of care in labeling, and preparing transactions.

If you found this guide helpful, please consider supporting myself or others involved in Bitcoin

- To me: bc1qg99fg656zflvfffvs7pgzje4rxe6sh8dw4p9nl
- To Casey of Ordinals: https://docs.ordinals.com/donate.html
- To Sparrow Wallet: https://sparrowwallet.com/donate/
- To Bitcoin Projects: https://opensats.org/