Setting up Ordinals for Bitcoin Inscriptions on MyNodeBTC

Table of Contents

Synopsis

Many Bitcoin node runners use MyNodeBTC on Raspberry Pi. This popular setup provides a web interface to enable and setup common Bitcoin applications and has minimal need for users to manage the internals.

The Ord program is currently strictly a command line tool, and it has some dependencies that rely on connecting to the Bitcoin service running on top of MyNodeBTC. This guide is intended to provide a MyNodeBTC node runner the information they need to get running with ordinals and be able to use the ord tool in general.

Prerequisites

Before going further, users of MyNodeBTC should have updated their node to version v0.3.07 or newer which includes the Bitcoin v24.0.1 release as well as the workaround patch for the Bitcoin wallet migration issue on that node.

Either use the access the command line on the node from the Linux Terminal button on the Settings page, or more preferably use a standard SSH client. For windows users you should be familiar with connecting to yoru node via SSH via PuTTy or via the tools from cygwin.

SSH in as the admin user and continue.

Review Bitcoin Configuration

To use Ordinals with Bitcoin, your Bitcoin node must be a full node. It cannot be operating in pruned mode. Review, and if necessary change the bitcoin configuration file as necessary. A typical setup of MyNodeBTC will not require any changes to the bitcoin.conf file.

NOTE: These settings can be potentially configured elsewhere via UI applications in MyNodeBTC. Ultimately the most important setting is that your Bitcoin server is running as a full node, not a pruned node.

1. Edit the bitcoin.conf configuration file

```
nano /mnt/hdd/mynode/bitcoin/.bitcoin.conf
```

2. Look through the file and make sure each of the following are set. If any of these are missing, add lines for them. They may not necessarily be grouped together:

```
server=1
listen=1
txindex=1
```

For an explanation of the bitcoin.conf settings, you may want to checkout Jameson Lopp's Bitcoin Core Config Generator tool at https://jlopp.github.io/bitcoin-core-config-generator/

3. Look through the file and see if there are any prune settings. If present, comment out the line by putting a # marker at the beginning of the line.

4. When finished making changes, save (press CTRL+O) and exit (press CTRL+X)

5. If you made changes, restart the Bitcoin Service.

```
sudo systemctl restart bitcoin
```

Installing and Using Tmux

Tmux is a terminal application that acts as a screen multiplexer. Within it you can have multiple terminals split with panes, and even multiple windows. But we're going to install it to help allow for long running processes to keep running.

```
sudo apt-get install tmux
tmux
```

You may want to print out or have handy the https://tmuxcheatsheet.com to keep keyboard combinations handy. The most important ones are

- List Bindings CTRL+b ?
  press q to exit list
- Create a vertical pane: CTRL+b "
  Create a horizontal pane: CTRL+b %
- Move between panes CTRL+b arrow-key
  Resize pane: CTRL+b CTRL+arrow-key
- Detach from session CTRL+b d

Programs running in a detached session will continue to run, even when you exit your SSH session or get disconnected. You can reattach to the session by launching tmux in attach mode

```
tmux attach
```

And type "exit" as you normally would to exit a terminal

Building Ord

At the time of this writing, prebuilt binaries for the Raspberry Pi were not available from the github releases page for the Ordinals project. https://github.com/casey/ord/releases

As such, your best option as a MyNodeBTC user is to build the tool yourself. This will also include the latest bugfixes, which is critical when it comes to the index that will be built later. This is explained in the steps below

1. As the admin user, install some required libraries and Rust

```
cd /tmp
sudo apt-get install libssl-dev git
curl --proto '=https' --tlsv1.3 https://sh.rustup.rs -sSf | sh
```

This step will take some time as rust and cargo tools are being prepared

2. Clone and build the ord tool

```
cd /mnt/hdd
sudo mkdir ordinals
sudo chown admin:admin ordinals
cd ordinals
git clone https://github.com/casey/ord.git
cd ord
source $HOME/.cargo/env
cargo build --release
```

The output of this will create a binary named “ord” in /mnt/hdd/ordinals/ord/target/release

Updating Ord

If in the future you want to update your ord client to the newest codebase, you can perform the following git commands and build again

```
cd /mnt/hdd/ordinals/ord
git checkout master
git fetch
git rebase origin/master
source $HOME/.cargo/env
cargo build --release
```

Configure Alias for Ord

Whenever you run ord, you will need to provide some key information including the data directory for the index, the bitcoin data directory, and the cookie file for accessing the bitcoin service via RPC calls.

To streamline this, we'll add an alias to your profile which will get loaded whenever you SSH in the future.

1. Edit your ~/.bashrc file

```
nano $HOME/.bashrc
```

2. Within this file at the very bottom, add the following line. All of this should be on a single line even if it appears to wrap within this document

```
alias ord='/mnt/hdd/ordinals/ord/target/release/ord --data-dir /mnt/hdd/ordinals --cookie-file /mnt/hdd/mynode/bitcoin/.cookie --bitcoin-data-dir /mnt/hdd/mynode/bitcoin'
```

3. Save (press CTRL+O) and exit (press CTRL+X) the file.

4. Source the file to load that alias into your current session

```
source ~/.bashrc
```

Now whenever you write a command with 'ord' it will effectively include all those common command line arguments for you simplifying what you need to type.

Test it out by checking the version

```
ord --version
```

Using Ord

With a fully synced Bitcoin node, and your Ord tool recently built, you're ready to use it. Before creating inscriptions, you will need to build a full index. On a raspberry pi, this process will take anywhere from a few days to over a week. As such, I recommend taking care of some other key things before indexing.

There is a good help system with ord. Every command you can add --help as an option to get output explaining their usage. Try it now

```
ord --help
```

Create a Wallet

You'll want to create a wallet. Ord uses Bitcoin Core as the backend for wallets and relies on the new descriptor wallets. By default, ord works with a wallet named 'ord'. If you want to use a different wallet, you can provide that on the command line by providing an additional argument denoting the name.

Just create a wallet named ord

```
ord wallet create
```

Specify a wallet name during creation. In this case, the wallet's name is "superwallet"

```
ord --wallet superwallet wallet create
```

In either case, you'll see output of 12 seed words that you'll want to write down to backup the wallet and allow for restoring it later. These same 12 seed words can be used to import into Sparrow Wallet on the desktop or other BIP39 compliant wallets.

Create a Receive Address

Whether you plan to create inscriptions or receive them, you'll need receive address(es). Each time you call this command the next receive address will be generated. It's worth noting that if you import the wallet into Sparrow, the receive addresses given here are actually change addresses. It doesn't matter as you have full access to them either way.

But generating a receive address now means you'll be able to send Bitcoin to it for inscribing while following later steps in the guide

```
ord wallet receive
```

A receive address will be displayed.

Begin Indexing

Now the lengthy process of indexing. Indexing, and updating the index will happen whenever you instruct ord to index, or if you perform a wallet operation that requires updated information about ordinals. For example, listing your inscriptions, or attempting to inscribe will cause it to want to index.

This is a time when its helpful to be running tmux. So if you aren't already, start that before indexing.

```
ord index
```

The index will begin processing through blocks, and this can take several days. The index file built will be stored in the location of the --data-dir indicated in our alias setting. In this case, you can see a file being built at /mnt/hdd/ordinals/index.redb.

Once its fully indexed, you'll be able to do all the commands for inscribing following the general guidance on the ordinals website: https://docs.ordinals.com/guides/inscriptions.html

Conclusion

Even though MyNodeBTC is WebUI centric, you can still use your node to work with tools on the command line. With this guide, you are able to build ord from source and update in the future. Eventually I anticipate a GUI wrapper around ord to make it easier to use, and binaries for the raspberry pi to be released through github. But even still, you're now more empowered to use your node as you see fit.

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

- To me: bc1qp0fam7thmrev0lme04ntacyjz4lxrzg69suvzt
- To Casey of Ordinals: https://docs.ordinals.com/donate.html
- To Open Source Projects: https://opensats.org/projects/