

Below you'll find a list of currently available applications and their compatibility with different operating systems.

- **hcd**: The node daemon, this command-line application handles block management and consensus.
- **hcwallet**: The wallet daemon, this command-line application handles address and transaction management.
- **hcctl**: The remote procedure call (RPC) client, this command-line application is used to control hcd and hcwallet via RPC commands.
- **hcGUI**: This application acts as a cross-platform GUI client as of R2.1.2.

	HCD	HCWALLET	HCCTL	HCGUI
Windows	✓	✓	✓	✓
macOS	✓	✓	✓	✓
Linux	✓	✓	✓	✓
Other UNIX	✓	✓	✓	✗

*'Other UNIX' includes various *BSDs and Solaris/illumos.

- **Note:** You might soon notice one of the many differences between HC and other popular cryptocurrencies: the wallet daemon and node daemon are separate. A lot of other coins run these functions together in a single daemon. For those who choose to use the command line interfaces, this means you must run hcd for full node functionality, and hcwallet to store your hc, create transactions and participate in Proof-of-Stake mining/voting.

The following are three methods that can be used to deploy hcd nodes:

1. **Download the Binary files**

- If you prefer not to use the hcd resource code, then you can deploy a hcd node by downloading the latest binary files on the [hcd release page](#). This method can be conducted on Linux, Windows and OSX.
- Take Linux as an example. First, download either the 32-bit or 64-bit binary files as needed. After downloading and extracting the files, open `linux_amd64_hcd` and `linux_amd64_hcctl` to see all executable binary files, including: `hcd` and `hcctl`.

2. Compile hcd from source code

- If you prefer to compile hcd directly from the source code, please download the latest version of this from our [github](#).
- Please refer to this [readme document](#) for the specific compilation method.
- After compiling, you will be able to get the executable binary files: `hcd` and `hcctl`.

3. Use DOCKER to deploy nodes

- Download the latest source code on Github (<https://github.com/HcashOrg/hcd>)

Compile Docker image

- `docker build -t hcashorg/hcd`

Start Hcd and name it hcd-1

- `docker run --name hcd-1 --rm -i -t hcashorg/hcd`

Run hcctl

- `docker exec hcd-1 hcctl getbestblock`

Run a HC node:

- You can run a hcd full node by using: `./hcd`
- To get more options, execute: `./hcd /help`
- To end the operation, please execute: `Ctrl+c`

- The default data path of the program is: `~/hcd/`. If the path is not empty during initialization, please clear the node and then run again (*ensure you backup before emptying*).
- All blocks need to be synchronized before initialization. The node running log path can be found at: `~/hcd/logs`.

Native RPC calls:

- When the node is running, native RPC calls can be implemented by interacting with `hcctl`.

For example:

```
oldclock@raven:~/hcd/cmd/hcctl$ ./hcctl getinfo {  
  "version": 2000300,  
  "protocolversion": 4,  
  "blocks": 21201,  
  "timeoffset": 0,  
  "connections": 2,  
  "proxy": "",  
  "difficulty": 127474315.32294893,  
  "testnet": false,  
  "relayfee": 0.001,  
  "errors": ""  
}
```

RPC command list:

- To obtain the RPC command list, please execute: `./hcctl help`
- To obtain instructions for RPC calls, please execute: `./hcctl help <RPCcmd>`.
- For example: `./hcctl help getinfo`