DigiByte Integration Guide

V1.3

DigiByte BLOCKCHAIN

www.promotedgb.com

Thank you for your interest in DigiByte

DigiByte is a 5-year old UXTO blockchain that started in 2014 with a focus on speed, scalability and security.

DigiByte is proudly not an ICO, and the small 0.5% pre-mine was given away completely to community members in full in the first 30 days of creation, in order to encourage adoption and full-node downloads.

DigiByte has no founders reward or block-fee, and although the founder is still actively engaged in DigiByte, any DigiByte held by him or the developers was obtained through purchasing on an Exchange, or mining on their own, just as any and all other users would. In addition, there is no individual that controls any significant portion of the circulating supply, such as the founder or developers. DigiByte is highly decentralized and fairly distributed.

DigiByte has a sound reputation both for its forward thinking and for its rapid responses to changing conditions. DigiByte was the first major UTXO blockchain to implement SegWit, several weeks ahead of both Litecoin / Bitcoin. DigiByte was also the first non-Bitcoin blockchain to implement a fix for CVE-2018-17144.

DigiByte is also the first major UTXO blockchain to implement the Dandelion++ privacy protocol with the 7.17 release.

DigiByte pioneered the DigiShield difficulty adjustment algorithm that is used in dozens of other blockchains, such as Bitcoin Cash, Ubiq, ZCash and more. DigiByte was also the first blockchain to switch from a single algorithm to multi-algo for increased security that 5x mining algorithms provides. DigiByte will also be pursuing ASIC-resistant algorithms throughout 2019, through an algorithm that changes every 10 days making ASIC creation pointless. Another solid first.

DigiByte mining is also highly distributed and decentralized, with all algorithms mining 20% of all blocks and the rewards distributed between them. This also further encourages the immediate processing of all transactions, due to other hardware vendors and types being used otherwise across all algorithms.

Finally, we want to welcome you to the global DigiByte community, and to thank you for taking the time to integrate with DigiByte. Our vast community is incredibly passionate about the DigiByte blockchain and we are sure that you will be too.

What are the standard ports for DigiByte integration?

DigiByte uses the following ports:

RPC Port: 14022
P2P Port: 12024
Testnet RPC: 14023
Testnet P2P: 12026

Both are TCP. While you do not need to have them publicly exposed / forwarded, doing-so does not present any security risk and allows your node to further contribute to the security of the DigiByte network by being discoverable by other nodes.

What should integrators be aware of regarding Dandelion++?

Dandelion++ is implemented in to DigiByte Core 7.17.2, alongside Odocrypt. Transactions are put into the “stempool” first, and the “mempool” second once the transaction has flowered.

If you are checking gettxout or similar for a transaction, please be aware it will not show immediately if you are using Dandelion.

You can alternatively disable dandelion using -disabledandelion=1 as a launch flag.

What should integrators be aware of regarding Odocrypt

Odocrypt is a new and unique hashing algorithm that morphs itself every 10 days, and will be replacing the Myr-Groestl algorithm from block 9,100,000 as part of DigiByte Core 7.17.2

Odocrypt was designed specifically to be ASIC resistant, due to these changes every 10 days, making the notion of creating an ASIC redundant as a company would effectively be creating a FPGA.

Odocrypt is already live on the DigiByte testnet and testing against it can be performed immediately.

Please keep in mind the change of epoch occurs daily at UTC+0 on testnet, where it occurs every 10 days on mainnet.

You can find more information about it at:
https://github.com/digibyte/digibyte/blob/7.17.2/src/crypto/odocrypt.cpp

Alternatively you can find more information on the wiki:
https://www.dgbwiki.com/index.php?title=Odocrypt

Do you have any good practices in integration with your wallet for deposits and withdrawals?

We recommend a minimum of **6** confirmations **for deposits** / **withdrawals** (90 seconds).
Notifications usually occur within 2-3 seconds for non-Dandelion transactions, with blocks being every 15 seconds.

If possible, use either Bech32 or SegWit addresses, and batch-process transactions.

Assuming that we will have 10 million users, how many wallets do you recommend we should have? Should we divide wallets / servers for different tasks, eg. address generating, deposit handling, withdrawals?

It genuinely depends on the capacity of the server, and the optimization of the software integrating with it. Look at your Bitcoin / Litecoin / Vertcoin server utilization for a good indication for you to base your DigiByte node from.

What parameters are recommended during installation / running a wallet?

You can connect to more than just the default number of nodes to ensure a geographical distribution, specifically adding the following to your digibyte.conf:

```
addnode=seed1.digibyte.io
addnode=seed2.digibyte.io
addnode=seed3.digibyte.io
addnode=seed.digibyte.io
addnode=seed.digibyteprojects.com
addnode=digihash.co
addnode=digiexplorer.info
addnode=seed.digibyteguide.com
addnode=seed-1.us.digibyteservers.io
```

However, in order to ensure a connection to as many nodes as possible as a "best practice", we would also suggest:

```
maxconnections=300
```

Most exchanges and services are also going to want to use the following:

```
txindex=1
```

What are the required hardware resources to run a single node?

A single node can happily seed on hardware as low as an *Intel Atom with 4GB RAM and 40GB HDD at present (2019)*, so the wallet will run on almost any Windows, Linux or OSX computer. However, this is probably less than ideal for an **Enterprise-grade production environment.**

For comparison, a quad-core 1.8Ghz Intel Atom with 4GB of RAM and will handle connections from approx 200 peers to the wallet, with the majority of that limitation being CPU-bound.
With this in mind we recommend you consider additional overheads for your own API calls, caching etc and where possible you should run digibyted on either bare-metal server, or a VM that has a solid-state drive.

Also, you should consider future growth of the blockchain, where now it is ~12GB, future growth requirements would suggest you actively monitor whatever server you run this on for drive-space usage etc.

Once your server is in production, you may also want to look at regularly clearing the debug.log file in ~/.digibyte/ . Please keep in mind that doing-so will severely limit your troubleshooting abilities down the line, should an issue arise.

Also, digibyted can and should always be run as *a non-privileged user* from within their home directory, and sudo access is not required at any time.

Where are 32-bit binaries?

We of 2018, 32-bit builds are no longer supported due to the number of blocks exceeding the addressable memory-space of 32-bit architecture.

DigiByte runs purely on 64-bit for the time being.

You can read more about this here: https://github.com/digibyte/digibyte/issues/144

What are practices to backup / restore wallet?

As per Litecoin / Bitcoin, *backing up the private keys or wallet.dat* is sufficient.
We recommend where possible you use a **2-of-3 multisig wallet.**

Are there any known restrictions or issues we should know?

Since early 2018, DigiByte has been in the process of changing newly generated wallet addresses from the "**D**" prefix to "**dgb1**" as part of bech32 support, and from "**3**" to "**S**" for **SegWit** addresses. Legacy address formats will continue to be supported indefinitely, so please allow for both.

As of **6.16.x**, DigiByte has implemented upstream Bitcoin Core 0.17 RPC formats. Common calls that exchanges / pools need to know about is that **getinfo** has been replaced by:

- **getblockchaininfo**
- **getnetworkinfo**
- **getwalletinfo**
- **getmininginfo**

In addition, **signrawtransaction** has been split in to two calls:

- **signrawtransactionwithkey**
- **signrawtransactionwithwallet**

signrawtransactionwithkey requires private keys to be passed in and does not use the wallet for any signing. **signrawtransactionwithwallet** uses the wallet to sign a raw transaction and does not have any parameters to take private keys.

We strongly advise also being aware of this where integrators are using the RPC calls directly, this call will no doubt be deprecated across other wallets going forward too as they bring themselves up to speed with the Bitcoin Core codebase.

Where existing products and services are using **signrawtransaction**, you should simply use **signrawtransactionwithwallet** in it’s place.

In addition, if you’ve been making use of the **ismine** value in **validateaddress**, you’ll want to instead call **getaddressinfo**, as the result is being returned there.

The **accounts** RPC call is also being changed, we recommend viewing the release notes from Bitcoin Core:
https://github.com/bitcoin/bitcoin/blob/master/doc/release-notes/release-notes-0.17.0.md

Do you have a sample digibyte.conf that you would recommend?

Sure:

```
# Place this config in the following path:
# ~/.digibyte/digibyte.conf
server=1
listen=1
daemon=1
txindex=1
rpcallowip=127.0.0.1
maxconnections=300
addnode=seed1.digibyte.io
addnode=seed2.digibyte.io
addnode=seed3.digibyte.io
addnode=seed.digibyte.io
addnode=seed.digibyteprojects.com
addnode=digihash.co
addnode=digiexplorer.info
addnode=seed.digibyteguide.com
addnode=seed-1.us.digibyteservers.io
```

Do you have any additional APIs we can use to integrate with the DigiByte Blockchain?

We recommend you run an Insight blockchain explorer service, then use the APIs documented here:
https://github.com/bitpay/insight-api

A quick setup example would be for you to create a new linux user, and run the following:

```
useradd digibyte # Create a new user for the service to run as — DO NOT RUN AS ROOT
su - digibyte
curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.8/install.sh
| bash
# Exit your terminal session and log in again to apply the profile changes
exit
su - digibyte
nvm ls-remote # Find the latest LTS
nvm install v8.10.0 # Install the latest LTS
npm install -g digibyte-node # install DigiByte Node
digibyte-node create explorer
cd explorer
digibyte-node install insight-digibyte-api
digibyte-node install insight-digibyte-ui
digibyte-node start
```

This will start up the **digibyte-node service** in the foreground of your terminal. We would recommend checking in **~/explorer/digibyte-node.json** to confirm the port it is running on.

The service should be listening on **TCP Port 3001**, which you can browse to **http://ip.address:3001/insight/**

You will want to add the suggested nodes described earlier in to **~/explorer/data/digibyte.conf**

PLEASE NOTE: This digibyte.conf is different from that used by the standard digibyte wallet.

Finally to run this as a background process, you may want to:

```
npm install pm2 -g
pm2 start digibyte-node -- start
```

What are the advantages of running a full node?

We recommend all exchanges / products run their own node they can use for API calls, especially if they are expecting any sort of decent volume. This can be run inside a virtualized environment without any additional configuration requirements.

Running your own node will allow the best possible performance for your product. Although there are hundreds of thousands of DigiByte nodes, very few run the Insight API on top of it for Blockchain integration. In addition, the DigiByte Foundation **was not an ICO and cannot sustain all of the server needs for the entire community directly.**

Do you have a running testnet?

Yes, please set:

testnet=1
In your digibyte.conf

The testnet operates on **TCP 12026.**

Who should we contact for additional technical support?

All of our developers and most of our community operate out of Telegram.

Or if your matter is for a broken package etc that requires a developer:
https://t.me/DigiByteDevelopers

In addition, the community member who provided this document to you should also be able to put you in touch with somebody who can assist with technical queries via email.

Should English not be your preferred language, please be sure to mention that, as our global community is very multi-lingual.

Can you confirm details for the DigiByte logo that we should be using?

You can download the DigiByte logo in a variety of formats from:

https://github.com/digibyte-core/digibyte-logos

Please ensure you are using the updated Logo which was released in October 2017, as this is different to what is on the first post of the BitcoinTalk announcement thread.

What is the download link for the official Wallet?

The Wallet can be downloaded from:
https://github.com/digibyte/digibyte/

You may build from the Master branch, as this is usually the latest stable-release branch. Changes are committed to their own branch, which is tagged for release, and then merged in to Master.

Any service should ideally follow best-practice of building from source, however the binaries are naturally available for convenience.

Who is the best contact for updates / what is the best way to keep informed about new releases?

We recommend subscribing to the GitHub repository:
https://github.com/digibyte/digibyte/releases
To remain informed of any updates.

You may also want to pass on any escalation / contact details of your own to whoever provided you with this document, in the case of any high priority issues such as CVE-2018-17144. There are members of the community who have previously contacted exchanges, pool operators and other service providers to inform them of such priority upgrades. With planned algorithm swaps occurring throughout 2019, remaining up-to-date should be a priority.

If you downloaded this document yourself, please reach out to the community through the Developer Channel on Telegram at https://t.me/DigiByteDevelopers