DigiByte

WWW.DIGBYTE.ORG

INTEGRATION GUIDE

v 2.0

REVISED BY:

DigiByte Alliance

LAST REVISION:

2023 June

TABLE OF CONTENTS

DigiByte: The Secure and Decentralized Blockchain for the Digital Age

DigiByte, created in 2013, is a prominent blockchain technology designed for fast, secure, and decentralized digital transactions. It is one of the most mature and scalable blockchains in existence and is also known for its fast transaction speeds and low fees. DigiByte did not have an ICO and was initially launched with only a small 0.5% premine which was given away to community members within the first 30 days in order to encourage adoption and full node downloads. There were no founders' rewards, block-fees, premined treasury or funds, nor were there any established corporate entities with paid employees. All developers and contributors are volunteers. DigiByte also has one of the largest and most active blockchain communities in the world.

Key Features of DigiByte

MultiAlgo: DigiByte utilizes a unique multi-algorithm mining system to ensure enhanced security and decentralization. Mining is also highly distributed which makes DigiByte one the most decentralized, pure PoW blockchains in the world. Unlike X16R, MultiAlgo implementation is not a selection of hashing algorithms in a pseudo-random manner, but rather each of the five algorithms actively compete against each other to mine 20% of all blocks. This also further encourages the immediate processing of all transactions due to other hardware vendors and types being used across all algorithms. The five algorithms supported are SHA256, Scrypt, Odocrypt, Skein, and Qubit.

DigiShield: DigiShield is an advanced difficulty adjustment algorithm that ensures fair mining and protects the network from malicious attacks. It achieves this protection by recalculating block difficulty between each block, allowing for a faster correction when a multi-pool begins or ceases contributing to DigiByte rather than recalculating once every 2 weeks as is the case with Bitcoin. The DigiShield code has been adopted by other blockchains, such as Ethereum, Zcash, Dogecoin, and many others.

SegWit: DigiByte was the first non-Bitcoin blockchain to implement the fix for CVE-2018-17144, enabling faster transaction processing and increased scalability. If a single transaction can be considered as 250 bytes, the DigiByte blockchain can theoretically handle between 266 and 1066 transactions per second. i.e 4000000 bytes SegWit scaling / 250 bytes tx size / 15 seconds block time = 1066 TPS with full SegWit usage. Combined with sub-penny transactions costs/fees, this makes DigiByte extremely scalable and affordable.

Dandelion: DigiByte is also the first major UTXO blockchain to implement the Dandelion privacy protocol with the 7.17 release. The purpose of Dandelion is to obscure the IP source for a transaction. When a node generates a transaction without Dandelion, it transmits that transaction to its peers with independent, exponential delays. This approach, known as diffusion in academia, allows network adversaries to link transactions to IP addresses. Dandelion mitigates this class of attacks by sending transactions over a randomly selected path before diffusion.

Digi-ID: Digi-ID is a secure, decentralized authentication system that eliminates the need for passwords and enhances user privacy. It is an authentication protocol that can be linked to the assets layer of the DigiByte blockchain. The architecture of DigiByte enables leveraging its blockchain to issue "Assets" by a "Trusted Third Party" that reflect digital identity credentials that can be issued in a permissioned manner and stored on the public decentralized DigiByte blockchain network.

DigiAssets: DigiAssets is a secure, scalable secondary layer on top of the global DigiByte blockchain that allows for the decentralized issuance of assets, tokens, smart contracts, digital identity, and much more. DigiAssets can be used to securely and cryptographically represent anything we find in the real world- from real-world assets such as real estate, airplanes, boats and cars, to scarce digital pieces of art and music. Signed documents such as wills, deeds, and purchase orders, medical bills, advertisement data and so on can be protected as DigiAssets.

DIGIBYTE TECHNICAL SPECIFICATIONS

Denominations

Plural	DigiBytes
Ticker Symbol	DGB
Currency Symbol	Ɗ (Unicode: U+018A)
Precision	10^{-8}

Subunits

mDGB (miliDigiByte)	1/1,000
µDGB (microDigiByte)	1/1,000,000
ɗSats (digiSatoshi, Digis)	1/100,000,000

Development

Author(s)	JaredTate, SmartArray, digicontributer, MentalCollatz
Implementations	DigiByte Core
Initial Release	v1.0 / January 10, 2014
Latest Release	v7.17.3 / May 12, 2021
Development Status	v8.22 RC2
Website	DigiByte.org

Ledger

Ledger Start	January 10, 2014
Genesis Block Hash	USA Today: 10/Jan/2014, Target: Data stolen from up to 110M customers
Ledger Type	Public, Decentralized, UTXO based, Multi-Algorithm
Timestamping Scheme	Proof-of-work (Partial hash inversion)
Hash Function	Five individual SHA256, Scrypt, Odocrypt, Skein & Qubit
Issuance Schedule	Decentralized (block reward) Initially Ɗ72,000 per block, 1% reduced every monthly
Block Time	15 seconds, (75 seconds per Algo)
Algorithm Block Share	20% Block Share per Algorithm (5 Algorithms x 20% for total Block Share)

Ledger (cont.)

Block Size and Capacity	Max block size 1MB. 1066 TPS with 4x SegWit scaling
Difficulty Retarget	Every 1 Block, 5 Separate Difficulties, 1 For each Mining Algo
Supply Limit	Ð21,000,000,000 (Supply limit reached in 2035)
Address Formats	D prefixed legacy addresses S prefixed p2sh SegWit compatible / MultiSig addresses dgb1 prefixed bech32 native SegWit addresses
Block Explorer(s)	digiexplorer.info, chainz.cryptoid.info, dgb.ccore.online

DIGIBYTE INTEGRATION GUIDELINES

Current source code:
https://github.com/DigiByte-Core/DigiByte/tree/release/v7.17.3

Current binaries (SIGNED):
https://github.com/DigiByte-Core/digibyte/releases/tag/v7.17.3

Full node docker image:
https://github.com/DigiByte-Core/digibyte-docker

APIs to integrate with the DigiByte Blockchain:
We recommend you run an Insight blockchain explorer service, then use the APIs documented at https://github.com/DigiByte-Core/insight-api.
A detailed setup guide is available on our Wiki, with step-by-step instructions available here: https://www.dgbwiki.com/index.php?title=Running_your_own_Insight_explorer

Magic Number:
The first 04 bytes of a block in the DigiByte Blockchain contains a magic number (constant) which serves as the network identifier.
Mainnet: 0xfac3b6da
Testnet: 0xfdc8bddd

Standard Ports for Integration (All TCP):
Mainnet RPC Port: tcp 14022
Mainnet P2P Port: tcp 12024
Testnet RPC Port: tcp 14023
Testnet P2P Port: tcp 12026

Derivation path for DigiByte:
DigiByte Legacy (D-prefix) addresses: m/44'/20'/0'/0/0
DigiByte SegWit/MultiSig (S-prefix) addresses: 49
DigiByte Bech32 (dgb1-prefix) addresses: 84

Hardware resources to run a single node:

At present, a single node can seed the DigiByte blockchain on an 64 bit CPU with at least 4GB RAM and 40GB HDD. The DigiByte Core wallet is supported on Windows, Linux, or OSX operating systems. While 4GB of RAM is currently the bare minimum recommended for a Linux VM, 8GB is the least that you should consider for deploying in a production environment and 16GB if you are utilizing additional services such as the Insight API services.

Expect a full synch to take at least 4 hours with a mid range CPU and an additional 24 hours if utilizing other services such as Insight APIs.

Also, you should consider the growth of the blockchain where it now consumes ~25GB (including indexes). Growth requirements would suggest you actively monitor drive space utilization.

Once your server is in production, you may also want to look at regularly clearing or log-rotating the ~/.digibyte/debug.log file. Please keep in mind that clearing our your log file, however, may limit your troubleshooting abilities down the road 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.

We recommend all exchanges, wallet products, and service providers run their own node especially for utilizing API calls or if there is expectation of significant volume. Nodes can also be run inside a virtualized environment.

Running your own node will allow the best possible performance for your product. Although there are hundreds of active DigiByte nodes, very few have integrated any 3rd party APIs such as the Insight API.

3rd parties that offer DigiByte infrastructure services:

DigiByte is supported at NOWNodes and GetBlock.

Recommended sample digibyte.conf for mainnet nodes and wallets:

Though customizing the default configuration file is unnecessary to get the node up and running, you can further customize the digibyte.conf:

```
# Place this config in the following path:
# ~/.digibyte/digibyte.conf
server=1
daemon=1
txindex=1
rpcallowip=127.0.0.1
maxconnections=300
addnode=seed.digibyte.io
addnode=seed.digibyteblockchain.org
addnode=seed.digibyte.help
addnode=seed.digibyte.link
addnode=digibyteblockexplorer.com
addnode=eu.digibyteseed.com
```

Recommended sample digibyte.conf for testnet nodes and wallets:

```
# Place this config in the following path:
# ~/.digibyte/digibyte.conf
server=1
daemon=1
txindex=1
testnet=1
rpcallowip=127.0.0.1
maxconnections=300
addnode=testseed.digibyteblockchain.org
addnode=testnet-1.us.digibyteservers.io
addnode=testnetseed.digibyte.help
addnode=testnetseed.digibyte.link
addnode=testnet.digibyteseed.com
```

BEST PRACTICES

RPC formats:

As of DigiByte Core 6.16.2, DigiByte has implemented upstream Bitcoin Core 0.17 pre-release RPC formats. Common calls that exchanges and pools should be aware of are *getinfo* has been replaced by:
getblockchaininfo, getnetworkinfo, getwalletinfo, andgetmininginfo

In addition, *signrawtransaction* has been split in to two calls: *signrawtransactionwithkey* and *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 most likely be deprecated across other wallets going forward as they get up to speed with the Bitcoin Core codebase. Where existing products and services are using *signrawtransaction*, you should simply use *signrawtransactionwithwallet* in its 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

Wallet Backup / Restore:

Much like Bitcoin, backing up the private keys or wallet.dat is sufficient, though care should be taken with encryption etc. We recommend where possible you use a 2-of-3 MultiSig or similar setup.

Legacy Address support:

Since early 2018, DigiByte has been in the process of changing newly generated wallet addresses from the legacy "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 capacity to send to both. As of DigiByte Core 8.x we will generate Bech32 addresses by default, so support is a priority.

Integration with your wallet for deposits and withdrawals:

We recommend a minimum of 8 confirmations for deposits and withdrawals. However, as you monitor the value of your transactions deposited, we recommend increasing the confirmations up to 50 for higher-value deposits in order to account for the SHA256 and Scrypt algorithms in the unlikely event of collusion from some of the larger mining pools that mine BTC or LTC for example.

Notifications usually occur within 1-2 seconds for non-Dandelion transactions, with block creation remaining unaffected at every 15 seconds.

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

Odocrypt Algorithm:

Odocrypt is a hashing algorithm that morphs itself every 10 days and has replaced the Myr-Groestl algorithm from block 9,100,000 as part of DigiByte Core 7.17.2.

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-Core/digibyte/blob/archive/7.17.2/src/crypto/odocrypt.cpp
Alternatively, you can find more information on the wiki: https://dgbwiki.com/index.php?title=Odocrypt

Dandelion support:

Dandelion was implemented in DigiByte Core 7.17.2, along with Odocrypt. With Dandelion, the transactions are put into the "stempool" then the "mempool" 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. However, we recommend only using this as a last resort.

Official DigiByte Logos and Guidelines:

Logos: https://github.com/DigiByte-Core/digibyte-logos
Guidelines: https://github.com/DigiByte-Core/digibyte-logos/blob/master/Logos%20Bitmap/Guide.png

DigiByte Dark Blue color code: #002352
DigiByte Light Blue color code: #0066cc

Please ensure you are writing DigiByte with capital D & B and that you are using the updated Logo described here, which was released in October 2017 as this is different to what is on the first post of the BitcoinTalk announcement thread. It is recommended that you use the Unicode character Ð (U+018A) when using DigiByte as a currency symbol.

QUESTIONS? CONTACT US.

Development Support:
While developer channels exist on Telegram and Discord, the best method of communicating directly with the core developers is through Gitter. Please communicate all inquiries in the DigiByte-Core/Community Gitter Channel.
https://app.gitter.im/#/room/#DigiByte-Core_community:gitter.im

Updates / New Releases:
We recommend subscribing to the GitHub repository to keep informed about all updates, releases, and any information pertaining to development.
https://github.com/DigiByte-Core/digibyte/releases

DigiByte Alliance:
If you have any questions about this documentation or wish to communicate to the DigiByte Alliance, you may reach us info@dgballiance.org or on Twitter @DGBAlliance.